



UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE GRADO EN INGENIERÍA INFORMÁTICA:

*Planificación Inteligente y
Contextualizada De Rutas De
Turismo*

Tutor: José María Álvarez Rodríguez

Autor: Marta Plaza Rodríguez

Junio 2016

AGRADECIMIENTOS

A toda mi familia, en especial a mis padres, por su apoyo incondicional durante todos estos años, por darme una educación, la oportunidad de estudiar a pesar de las dificultades económicas y ante todo siempre creer en mí.

A mi hermano, gracias por ayudarme cada día a cruzar con firmeza el camino de la superación y ser una guía en mi vida.

En especial a mi abuelo Juan, que a pesar de su reciente fallecimiento, y sabiendo que ya jamás encontraré la forma de agradecer su constante apoyo y confianza, me llena de gran satisfacción dedicarle dicho Trabajo Fin de Grado, siempre te recordaré.

A mis amigas y a Pablo, por haber estado junto a mí en los momentos más difíciles y cuando más lo necesitaba, a quienes jamás encontraré la forma de agradecer el cariño, comprensión y apoyo brindado hasta el momento.

A mis compañeros de clase, con quienes he compartido esta importante etapa de mi vida. Por unos años universitarios inolvidables, que a pesar de aquellas situaciones de estrés sufridas durante las prácticas en las famosas Fs, siempre hemos conseguido superarlas, y gracias por todos aquellos momentos tan buenos que me habéis hecho pasar, os echaré de menos.

En último lugar y no menos importante, darle las gracias a mi tutor José María, por guiarme durante el desarrollo del proyecto y darme todos sus ánimos y fuerza para conseguirlo.

“Tu tiempo es limitado, de modo que no lo malgastes viviendo la vida de alguien distinto. No quedes atrapado en el dogma, que es vivir como otros piensan que deberías vivir. No dejes que los ruidos de las opiniones de los demás acallen tu propia voz interior. Y, lo que es más importante, ten el coraje para hacer lo que te dicen tu corazón y tu intuición”

Steve Jobs.

ABSTRACT

First of all, this document specifies the main objectives and structure that will continue throughout it in order to give a general idea about the contents of the report, before an explanation in greater depth about the project. Below, the main reasons of selection and realization of this final degree project are presented.

The reasons which motivated me and influenced to carry out this project were, mainly, to have the opportunity to learn how to develop my own web application, taking into account the knowledge received during my educative training. On the other hand, there are several reasons why I decided to do this implementation based on the tourism sector. This is justified by a historical summary which mentions the main events that encouraged its creation and certain features to consider.

In the history of humanity, there has always been a social desire to escape temporarily from everything. The main reason would be leaving the everyday environment without worrying too much about where to go. Preferably, it's always better to choose a more pleasant environment than the one that envelops the own daily routine. Therefore, in the case of the tourism, this reason constitutes the base of the wish to travel and includes the creation of a necessity.

From an academic point of view, this need for travel deals with different disciplines, so that different theories and models have been developed concerning this purpose. In 1943 Abraham Maslow published his motivational needs model following a hierarchical structure organized in five levels:

1. Physiological needs (at the bottom of the pyramid).
2. Physical and mental security needs.
3. Social belonging, affection and relationship.
4. Esteem and social recognition needs.
5. Self-actualization needs and personal development (at the top of the pyramid).

This theory has had more applications in the field of study of the tourism sector. Among others, there is the theory which was made by Pearce -that was based on the model proposed by Maslow-. This new model proposed an adaptation incorporating the tourist experience factor. In this way, the levels that exist in the field of the tourist motivation will be arranged in the following way:

1. Need for relaxation (Take a break/activity).
2. Need for stimulation (security / strong emotions).
3. Social needs (of family and intimate relationships of friendship).
4. Self-esteem needs (personal, cultural, historical and environmental development).
5. Self-fulfillment needs (search for the happiness).

In this way, it is important to point out that this model is based on certain western aspects conversely to many parts of the world where the ultimate goal would be the community service rather than the needs for individual self-fulfillment.

Therefore, the tourism offers a release from the repression produced by work and the rest of the obligations, getting some distance from the domestic environment and creating an inner desire to acquire new knowledges.

With reference to the pyramidal models of Pearce and Maslow, it is important to mention that these were those that motivated the fact that the web application was based on the needs of their higher levels. In this way, these levels focus on the desire to learn new things, to experience different cultures, discover yourself, which is the part of the personal search that the tourist pretends to find in his destination.

According to the previously mentioned aspects, it seemed to me an important and essential area to bring technology to. As we have seen, this represents a need and even, it constitutes a right which contributes, at the same time, to the development of the countries and regions.

Regarding the main goals of the project, it is necessary to say that the main objective of the project is based on the design and implementation of a web application focused on the tourism sector, as we have mentioned previously.

This application will try to show the users the metadata belonging to a word (either a person, place, organization or event, among others) introduced by themselves in the searcher and through queries to the databases called DBpedia and Google Places. Once they have obtained the metadata (which are the characteristics associated with that issue that the users searched), they will select one or more of them and that will allow the creation of one or more end-points, also using queries to the database of Foursquare.

In this way, those points of interest obtained by the users will be shown on a map and there will be created routes whose purpose will be to complete those end-points. It is essential to mention that the creation of routes can be done from one or several features emerging from different searches of the users, that is to say, the route may be formed from the search of different characteristics that not necessarily have something in common.

Consequently, it is tried to improve the users' satisfaction and experience approaching the technology by means of the visit of different places from its habitual environment. This constitutes a way of entertainment for the customer and, at the same time, it can strengthen knowledge about historical figures, events, museums and cultural events which are featured, among others, in relation to each point of interest.

Moreover, the main aim is to create the user experience through the digital world by routes fulfillment, so that it will provide the possibility of creating an environment full of challenges and motivation when it comes to learning the general culture of any region, like it were a game, but, in this case, concerning real life. Also, it allows us to promote the tourism, something which will contribute to an economic profit from the places visited by the user of the application.

It is necessary to say the characteristics which must be met:

- It must be easy to use
- It must be able to provide agile and instant information
- It must be practical and it must help you to organize yourself.

Tourism and technology go hand in hand. The fast and continuous development of new technologies help the tourist activity to offer new experiences and tools for the best development of tourism.

In an extremely connected and interrelated world today, the way in which companies or governments may go adding these technologies and apply them to the tourist activity, will mark differences in terms of the quality of services and, especially, the way that they can be known in the world through them.

Currently, the tourism industry is facing a new environment derived from the transformations that this industry has undergone. This aspect has come to modify the trends of the sector before the demands of a market that requires concepts more personalized and flexible from the product or tourist service. In view of this situation, companies need to guide and focus its business management model on the needs of its customers to face the changes and the strategic challenges that they have planned. In this context, they need more quality information about the developed activities, the market in which they operate and the agents that they are in contact with.

The mentioned aspects, make the information and knowledge about the customer a key asset for the management of the tourist activity that may, in addition, be critical to improve the level of competitiveness of the companies in this sector. Certainly, it is essential to provide the best instruments of systems/information technologies to obtain valuable information that serves as support to improve the management of business activity and the quality of the customer's service.

Given this situation, to have precise, opportune, and excellent information is the base for the company to be able to be successful in the difficult task of satisfying the tourism demand.

On the other hand, the following characteristics about several web applications of the tourism sector are described in the order to carry out a comparative analysis according to the web application which will be developed:

- TripAdvisor:
 - It allows users to read reviews about people who already visited selected destination.
 - It shows information about any desired destination.
 - It makes possible to access to the purchase of the tickets of a plane, hotels booking or holidays rentals.
 - You can add your own opinions, videos, and photos about any destination.
 - Travelers can see on the map the score and opinions of other travelers to compare several options.
- TouristEye:
 - It collects ideas for your trips and getaways.
 - It recommends sites around your destination of interest.
 - Each destination has photos, comments and its location on a map.
 - It shows the evaluation of the destinations by other travelers who already visited it.

- Minube:
 - If you even don't know the destination to travel, it allows you to be inspired by thematic lists of different places.
 - It gives you the possibility to create your own travel guide of your desired destination which will include:
 - Places to see.
 - Establishments to eat.
 - Places to stay.
 - It provides the functionality to add photos, multimedia and files to your user profile.
 - It shows interest points about your destination in the map.
- Tripit:
 - It allows the user to make a personalized planning of its trip adding different activities associated with each day that the user stays in his destination.
 - It allows you to add or delete plans manually.
 - It synchronizes your calendar with travel plans.
 - It makes possible to share the already made tourist routes and other information between the different friends who have an account in Tripit.
- The Culture Trip:
 - It allows you to discover the information about the typical food, art, cinema, literature, trips, among other culture aspects of each country. This cultural information has its origin in the stories and recommendations from each of the 15 associated branches.
 - It allows the search about the cultural information of a specific location.
 - The already made searches can be made by a cultural thematic: Art, Food, Travel, Music, Film, Literature, Design, History and Other.

After the study of the different aspects about the similar web applications, we can observe that it is necessary to include in the project several aspects in common with them that were considered useful, but in other cases some of them were irrelevant in relation with the pursued objective.

Therefore, the functionalities that are considered more important for the implementation process are the following ones:

- **Location of POIs in the map:** regarding this functionality, all the web applications except The Culture Trip fulfills it. It was considered an important aspect since it allows to establish a relation between technology and the real world allowing the user to associate the remoteness or proximity of each point of interest (POI) that you want to visit.
- **Search by entity:** as for this this functionality, we can say that it has not been implemented by any similar web application, so that it was considered an improvement: it do not only allows to search by location, but also to search by entity (it could be a person, place, organization, event or others) in order to expand the search area to plan the desired trip. In this way, in the case of searching by an entity the web

application will obtain a set of characteristics as a result. These characteristics will be selected by the user according to his or her interest, from which they will extract locations in the process of creating routes.

- **Add / delete POIs:** in relation to this aspect, the similar applications that fulfilled it could add points of interest around the searched destination. However, it was decided to make an improvement of this functionality by allowing you to avoid adding or deleting POIs that necessarily are found in the surroundings of a searched destination and substituting it with the possibility of adding or deleting these characteristics associated with the requested entity (person, place, organization or event, among others) or that may not have any relation. That is to say, instead of that the points of interest having relation because of the distance for the searched destination, it was intended that they had relation for the searched entity or without necessarily having a relation between them.
- **To show multimedia information from each POI:** This functionality has been adopted by the most of the similar web applications and was considered useful for the user to have more knowledge about the places that he will visit during his trip. This information is constituted by the image of the place, description, source search from which it comes, name, address, among others.
- **Create route automatically:** Such functionality is one of the key points of the web application. As we can see, in similar applications it is not implemented because not all of them can create routes as such, just some of them can create manually personalized routes including activities or points of interest around the desired destination making a planning with the visits during the trip. However, there is a clear need to create routes with an automatic planning.

In this way, in the process of automatic creation of routes, the user will have to do the only task of adding or deleting those characteristics that are associated with the searched entity and the web application will obtain automatically a route with the points of interest associated to these features selected by the user and organized with the criterion of the shortest distance between them, without the necessity of planning manually the trips.

- **Evaluation of generated route:** As we have seen in most of the similar applications, it is given the possibility of making an assessment about a visited place, however, in the case of the applications that have the possibility of generating routes manually neither have the possibility to value the complete route. In this way, an improvement that it is a good idea to include is the evaluation of the generated routes.

After the analysis of the similar web applications, it is necessary to continue with the phase in the development of the project, which is called Analysis. In this process, it will be explained what is expected of the system which allows to identify those needs of the software product to develop, i.e., the required functionality of the web application that will constitute the project. To do this, firstly I will specify the user requirements made in parallel with the use case diagrams, which are used to specify the communication and the behavior of a system by means of its interaction with the users and/or other systems. Then, from these user requirements, it will be extracted the software requirements.

It is common to see the existence of many differences between the customer and the developer because in many occasions it is difficult to satisfy him or her. Therefore, there is a clear need to take into account the user during all the process of development of the project and, for that reason, it has been necessary to make the user requirements specification, in order to determinate the high-level abstract requirements.

However, the user requirements describe in a natural language the functional and not functional requirements that are understood by the users of the system that they do not have a deep technical knowledge.

Therefore, the user requirements extracted are the following:

- The user will be able to search any type of entity.
- The searched entity can be a person, location, organization event or other.
- The user will be able to visualize characteristics in relation to the searched entity.
- The user can select / discard the available characteristics according to the type of entity.
- The user can create routes from the selected characteristics from him or her.
- The user will be able to observe on a map the locations associated with the characteristics that he selected.
- The user must indicate the point of origin where he wants to start the route.
- The user will be able to visualize the order in which he or she must visit points of interest that constitute the already created route.
- The user will be able to value every created route.
- The user can click on each location on the map and visualize information in relation to it.
- The system will be able to show the characteristics obtained after a search in Spanish or English.
- The system displays de interface in Spanish.
- The web application will be adaptive to any device that uses it.

Also, it is important to explain the specific case of use. The case of use will show the sequence of interactions that take place between a system and its actors in response to an event.

In base or that, the cases of uses are defined below and it is necessary to say that the actor of these scenarios is the user:

- *Searching by entity*: The user pretends to obtain a set of characteristics concerning the searched entity.
- *Selection characteristics*: The user has to select some characteristics regarding the searched entity.
- *Discard characteristics*: The user has the possibility to discard the characteristics that were selected by him.

- *Create a route*: Create a route through the characteristics selected by the user.
- *Value a route*: Score previously created route.

After the detail of the user requirements and the use of case, it is possible to specify the necessary requirements to the later phases of the development of the project. These requirements constitute the fundamental base for the functioning of the web application and they are specified according to the guidelines given by the standard ANSI/IEEE 830, 1998.

Then, the next phase in the development of the project is called Design. According to Pressman: "The design of the software is really a process of many steps but that they are classified inside one himself". In this way, to carry out the definition of the design phase of the project the model that will be used is called "*Model 4+1*".

The *model 4+1* was designed by Philippe Kruchten, and it was used to describe the architecture of software systems, based on the use of multiple points of view.

A view is no more than a representation of the whole system from a certain perspective, and a point of view is defined as a set of rules (or regulations) to perform and understand the views.

Therefore, what Kruchten proposes is that a software system must be documented and show (as it is proposed in the standard IEEE 1471-2000) with 4 different views and these 4 views are to be interrelated with a view more, which is the view called "+1". These 4 views were coined by Kruchten as:

- Logical view.
- Development view.
- Processes view.
- Physical view.

And "+1 view", that has the function of relating the 4 view mentioned. It is called *view of scenario*.

Each of these views have to show all the architecture of the software system, but each one of them has to be documented in a different form and they have to show different aspects of the software system.

The different views of this model are shown below:

- *Logical view*: the logical view is focused on the structure and functionality of the system. It represents the functionality that is provided to the final users, i.e., it produces a representation about what the system has to do, and the functions and services that it offers. To do this, it will make use of a class diagram. In the case of this project, the types that the diagram has are the following: *SearchAppServ*, *POIDAO*, *POIROUTE*, *SPARQLPOIDAO*, *GooglePlacesDAImpl*, *FoursquareDAImpl*, *ENTITY*, *CONTEXT*, *POI*, *RoutesPOIROUTE*, *CONTEXTROUTE*, *Route*, *AspectsRoute*, *EntityRoute* and *ValueRouteRequest*.

Later on, this classes are explained with more detail in the present document.

- *Development view*: this view shows the system in the perspective of the programmer and it is responsible for the software management, i.e., it shows how the system is divided. To do this, it will make use of a components diagram and a packages diagram.
 - *Components diagram*: we can observe the different components in the web application and the relations between them. The components that appear in the diagram are: In-Tour view, In-Tour Web and the external services that the application uses: Google Places, DBpedia and Foursquare. This diagram follows *the Customer-Server Architecture*.
 - *Packages diagram*: it represents the dependencies between the packages that constitute a model, i.e., it shows how the system is divided in logical groupings and the dependencies between them. The package in this diagram will be: es.uc3m.intour.service, es.uc3m.intour.business, es.uc3m.intour.appserv, es.uc3m.intour.dao and es.uc3m.intour.to.

Later on, these diagrams are explained more in depth in this document.

- *Processes view*: this view takes on the dynamic aspects of the system, explains the process of the system and how they communicate. It focuses on the behavior of the system during the execution time and it considers aspects of concurrency, distribution, performance, scalability, among others. In this way, that view is represented by the sequence diagrams. In the case of In-Tour Web, later on these diagrams are explained with more detail there have been made the following ones: the sequence diagram for the search by entity, the sequence diagrams for the generating routes and the sequence diagram to value the routes.
- *Physical view*: this view describes the system from the point of view of a systems engineer. It is related to the topology of software components at the physical layer, as well as the physical connection between the components. It is represented by a deployment diagram.
- *+1 View*: this view constitutes the description of the architecture's system. It is represented by the cases of use defined in the analysis phase.

Finally, I will expose the conclusions that are extracted from the project and the future lines:

The designed and developed system is the beginning of a future engine of recommendations. It will help the user to schedule the routes and paths he or she wants to carry out during travels in a smarter and dynamic way.

Its development raised from the need of a strategy to create routes for people. Nowadays, the tourists are better informed and have more tools to find the different destinations and their dynamics. So, this behavior is changing day to day and the destinations are more different than years ago due to the fact that currently borders are limitless.

This tool pretends to give the user an opportunity to manage his or her travel faster and simpler, so for me was a pleasure to contribute to the tourism to provide any user the possibility to acquire an interesting knowledge about a visit to a place in the routes created by the tool.

The main aim of this application was implementing a web application. This application shows the user metadata related to a word (this word could be a person, a place, an organization, an

event or other) introduced in the searcher. From this method the user gets the points of interest related to the topic that he has written before, giving him or her a route to complete. Finally, to achieve our goal, this system has these main features:

- **Search by entity:** the system not only allow to search by a location, it lets the users search by place, people, organization, event or other key concepts. The result returned will have some features that the user can choose or discard. This last concept lets the users personalize their desired travel. In other similar web applications this functionality does not exist.
- **Add/Remove features:** the system lets the users add or remove any feature related to the entity searched (people, place, organization, event or other) and they are able to choose other features obtained from previous searches as well. The target in this case, is to make a combination of features from different searches with the goal of getting more culture, historic, etc. knowledge while the user is having fun with his or her trip. In other similar web applications, this functionality does not exist, because they only let the users add points of interest close to them, but never related to a topic or a huge route.
- **The system can schedule automatic routes:** in this way, the user has the only task of adding or removing the not desired features and the web application automatically gets the route formed by the features chosen by the user. These points of interest will be returned to the user organized by the criteria of short distance, in this case the user has not anything to do to manage or schedule the order of the places. It is a key functionality of the developed application, because in other similar applications they have the following cases of study:
 - The user only gets one result related to the concrete destination.
 - The user only would be able to add close places to his scheduled.
 - The user only has the possibility to create manually the route that he or she wants to do.
 - The order is managed manually by the user, and it is never done automatically.

In this way, this functionality is a really big benefit as it lets the user do less effort to manage his or her trips.

- **Map location places:** The system will show the users in the map the point of interest related to the features deleted by them.
- **Display multimedia POI's information:** The system will display the user the multimedia information of any place with just one click. The application would display the following data:
 - Name of the feature from which the point of interest comes.
 - Latitude.
 - Longitude.
 - Address of point of interest.
 - Original source of the search: DBpedia, Google Places or Foursquare.
 - Representative image of the place.

In spite of this functionality, it is not new in similar applications. It was considered of really great use to understand and know better each point of interest.

- **Score of generated route:** The system lets score the routes without having to score each of the points of interest contained in the route. Other applications give the possibility of scoring only one place but never a whole route.

The problems and difficulties emerged were, -almost all of them-, related to the setup of the libraries used for the communication with third party systems, like databases. In other hand, also we have problems during the data transferring from the front layer to the back one. All of these problems are resolved.

Definitely, the system meets all requirements specified from the beginning but is necessary to mention that the key aspect of the application is that it can connect with other systems and get information from them and improve the routes and the order of the point of interest. The main goal of this project is not to be better than the databases used (DBpedia, Google places or Foursquare), but it is to join a lot of information and produce a really good experience to the user with the collected data from different sources.

So, bearing in mind the previous points, like any system, it could be improved in different ways, thus, my proposals to do so, are:

- ***The compliance of commitment:*** this improvement consists of the adaptation of the application like if it were a game, but adjusted to real life. In this way, down by this developed web application it could be included the concession of commitment after every route completed by users. This added function will turn into a situation in which the user will get to a point of interests of the created route. So, the user will have the option to indicate that he or she has completed one of the places of the route. Therefore, the attainment led to compare the geographic location of the user through his geolocation with the latitude and length of the point of interests of the route which the user, supposedly, has just visited.
- ***Social network:*** this improvement will led users to have a profile in which they could show those visited places through the publication of pictures or videos. They also could chat via online and share comments in each created route, and much more actions. These functions are used by almost most of the similar web pages that were previously analyzed. As a result, it is considered as an important point to be included.
- ***Implementation in IOS and Android:*** in case of the application is used for a considerable number of users, it would be important to carry out it with the running of the In-Tour Web through an application for mobiles, due to the fact that it would be easier than a web page mobile. Therefore, the camera function or the geolocation will be more leveraged.
- ***Cost of route:*** other of the important improvements that could be included consists on the calculation of the approximated cost of the future travel. As a matter of fact, this will led the user to choose the cheapest tourist planning.
- ***Recommendations:*** as it has been mentioned before, it is possible to have an efficient use of the scores of routes that were stored. In this way, every score was stored in an anonymous method. So that, this better proposal will consist of using this information which establishes entrance pieces of information of an algorithm of recommendation –of type collaborative filtrated item-item- in order to create recommendations that are showed to the user and so, help him to decide about his travels.

ÍNDICE DE CONTENIDOS

Índice de ilustraciones.....	19
Índice de Tablas.....	20
1 Introducción	23
1.1 Motivación	23
1.2 Objetivos	25
1.3 Estructura del documento.....	25
2 Estado del arte	27
2.1 Aplicaciones similares	28
2.1.1 TripAdvisor	28
2.1.2 TouristEye.....	29
2.1.3 Minube.	30
2.1.4 Triplt.....	32
2.1.5 The Culture Trip.....	33
2.2 Análisis comparativo	34
3 Análisis.....	37
3.1 Requisitos de usuario.	37
3.2 Casos de uso.	41
3.2.1 Diagrama de casos de uso.	41
3.2.2 Actores	42
3.2.3 Descripción de los casos de uso	42
3.3 Requisitos de software.....	46
3.3.1 Propósito	46
3.3.2 Alcance	46
3.3.3 Personal involucrado.....	47
3.3.4 Descripción General	47
3.3.5 Especificación de Requisitos	55
3.4 Matriz de trazabilidad.	64
4 Diseño.....	65
4.1 Vista lógica	66
4.1.1 Descripción General del Diagrama de Clases.....	67
4.1.2 Tarjetas CRC	70
4.2 Vista de desarrollo.....	80
4.2.1 Diagrama de Componentes.....	80
4.2.2 Diagrama de paquetes	83

4.3	Vista de procesos	86
4.3.1	Diagrama de secuencia: Búsqueda por entidad.....	86
4.3.2	Diagrama de secuencia: Generación de rutas.....	87
4.3.3	Diagrama de secuencia: Valoración de rutas	89
4.4	Vista física.....	90
4.5	Vista +1: escenarios.....	91
4.6	Entorno tecnológico seleccionado	92
4.6.1	Java	92
4.6.2	Servicio Web: REST	93
4.6.3	Maven.....	95
4.6.4	Git	96
5	Pruebas realizadas.....	97
5.1	Pruebas unitarias.....	98
5.2	Pruebas de integración	100
5.3	Pruebas de usabilidad y accesibilidad	103
5.4	Resultado de las pruebas.	105
5.5	Métricas de código	106
6	Planificación y presupuesto	111
6.1	Planificación del proyecto	111
6.2	Presupuesto del proyecto	114
6.2.1	Costes de personal	114
6.2.2	Costes del material.....	115
6.2.3	Costes indirectos	116
6.2.4	Coste total del proyecto	117
7	Ética y marco legal.....	119
8	Entorno socio-económico	121
9	Conclusiones y trabajo futuro	125
10	Bibliografía	129
11	Glosario: Acrónimos, definiciones y abreviaturas.....	131

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Modelo Piramidal de Maslow [1]	23
Ilustración 2: Modelo Piramidal de Pearce [2]	24
Ilustración 3: Aplicación web TripAdvisor [3]	28
Ilustración 4: Planificación de viaje en TripAdvisor [3]	29
Ilustración 5: Aplicación web TouristEye [4]	29
Ilustración 6: Agregación de planes con TouristEye [4]	30
Ilustración 7: Aplicación web minube [5]	31
Ilustración 8: Puntos de interés en el destino con minube [5]	31
Ilustración 9: Fotos del lugar destino con minube [5]	31
Ilustración 10: Aplicación web Triplt [6]	32
Ilustración 11: Planificación de actividades con Triplt [6]	32
Ilustración 12: Aplicación web The Culture Trip [7]	33
Ilustración 13: Diagrama de casos de uso	41
Ilustración 14: Diagrama In-Tour Web con otros sistemas	49
Ilustración 15: Plantilla Escenarios de Perfiles de Usuario	50
Ilustración 16: Modelo 4+1	65
Ilustración 17: Diagrama de clases	66
Ilustración 18: Arquitectura Cliente-Servidor en 3 capas [9]	81
Ilustración 19: Diagrama de componentes	83
Ilustración 20: Diagrama de secuencia-Búsqueda por entidad	86
Ilustración 21: Diagrama de secuencia-Obtención de localizaciones	87
Ilustración 22: Diagrama de secuencia-Generación de rutas	88
Ilustración 23: Diagrama de secuencia-Valoración de Rutas	89
Ilustración 24: Diagrama de despliegue	90
Ilustración 25: Logo Java [10]	92
Ilustración 26: Logo WebServices REST [11]	94
Ilustración 27: Logo Maven [12]	95
Ilustración 28: Logo Git [13]	96
Ilustración 29: Diagrama de Gantt	112
Ilustración 30: Crecimiento anual del PIB turístico [21]	122
Ilustración 31: Ingresos por turismo interacional 2014 [21]	122

ÍNDICE DE TABLAS

Tabla 1: Análisis comparativo de aplicaciones web similares	34
Tabla 2: Plantilla definición requisitos de usuario	37
Tabla 3: Búsquedas RU-01.....	38
Tabla 4: Tipo de Búsquedas RU-02.....	38
Tabla 5: Caracterización entidad RU-03.....	38
Tabla 6: Características seleccionadas/descartadas RU-4	38
Tabla 7: Generación ruta RU-05.....	39
Tabla 8: Localizaciones en el mapa RU-06	39
Tabla 9: Punto de Origen RU-07.....	39
Tabla 10: Completar ruta RU-08	39
Tabla 11: Valoración ruta RU-09	40
Tabla 12: Información por localización RU-10	40
Tabla 13: Idioma de las características RU-11.....	40
Tabla 14: Idioma Interfaz RU-12.....	40
Tabla 15: Interfaz adaptativa RU-13.....	41
Tabla 16: Plantilla definición casos de uso.....	42
Tabla 17: CU-01 Búsqueda por entidad	43
Tabla 18: CU-02 Seleccionar características.....	43
Tabla 19: CU-03 Descartar características	44
Tabla 20: CU-04 Generar ruta	45
Tabla 21: CU-05 Valorar ruta.....	45
Tabla 22: Ficha descriptiva José María Álvarez Rodríguez	47
Tabla 23: Ficha descriptiva Marta Plaza Rodríguez.....	47
Tabla 24: Escenario Perfiles Usuario 01	51
Tabla 25: Escenario Perfiles Usuario 02	52
Tabla 26: Escenario Perfiles Usuario 03	53
Tabla 27: Escenario Perfiles Usuario 04	54
Tabla 28: Plantilla de requisito de software.....	55
Tabla 29: Requisito RSF-01.....	56
Tabla 30: Requisito RSF-02.....	56
Tabla 31: Requisito RSF-03.....	57
Tabla 32: Requisito RSF-04.....	57
Tabla 33: Requisito RSF-05.....	57
Tabla 34: Requisito RSF-06.....	58
Tabla 35: Requisito RSF-07.....	58
Tabla 36: Requisito RSF-08.....	58
Tabla 37: Requisito RSF-09.....	59
Tabla 38: Requisito RSF-10.....	59
Tabla 39: Requisito RSF-11.....	59
Tabla 40: Requisito RSF-12.....	60
Tabla 41: Requisito RSF-13.....	60
Tabla 42: Requisito RSF-14.....	60
Tabla 43: Requisito RSF-15.....	61
Tabla 44: Requisito RSF-16.....	61
Tabla 45: Requisito RSNF-01	61
Tabla 46: Requisito RSNF-02	62

Tabla 47: Requisito RSNF-03	62
Tabla 48: Requisito RSNF-04	62
Tabla 49: Requisito RSNF-05	63
Tabla 50: Requisito RSNF-06	63
Tabla 51: Matriz de Trazabilidad Requisitos Funcionales vs Casos de Uso.....	64
Tabla 52: Plantilla Tarjetas CRC.....	70
Tabla 53: Tarjeta CRC SearchAppServ	71
Tabla 54: Tarjeta CRC POIDAO	72
Tabla 55: Tarjeta CRC SPARQLPOIDAO.....	72
Tabla 56: Tarjeta CRC GooglePlacesDAImpl.....	73
Tabla 57: Tarjeta CRC FoursquareDAImpl.....	74
Tabla 58: Tarjeta CRC CONTEXT	74
Tabla 59: Tarjeta CRC ENTITY	75
Tabla 60: Tarjeta CRC POI.....	75
Tabla 61: Tarjeta CRC POIRROUTE	76
Tabla 62: Tarjeta CRC RoutesPOIRROUTE	77
Tabla 63: Tarjeta CRC CONTEXTROUTE.....	78
Tabla 64: Tarjeta CRC RouteRequest	78
Tabla 65: Tarjeta CRC Route.....	78
Tabla 66: Tarjeta CRC AspectsRoute	79
Tabla 67: Tarjeta CRC ValueRouteRequest	79
Tabla 68: Tarjeta CRC EntityRoute	79
Tabla 69: Diagrama de componentes	80
Tabla 70: Plantilla de pruebas	97
Tabla 71: Prueba unitaria UNI-01.....	98
Tabla 72: Prueba unitaria UNI-02.....	98
Tabla 73: Prueba unitaria UNI-03.....	98
Tabla 74: Prueba unitaria UNI-04.....	99
Tabla 75: Prueba unitaria UNI-05.....	99
Tabla 76: Prueba unitaria UNI-06.....	99
Tabla 77: Prueba unitaria UNI-07.....	100
Tabla 78: Prueba de integración INT-01.....	100
Tabla 79: Prueba de integración INT-02.....	101
Tabla 80: Prueba de integración INT-03.....	101
Tabla 81: Prueba de integración INT-04.....	101
Tabla 82: Prueba de integración INT-05.....	102
Tabla 83: Prueba de integración INT-06.....	102
Tabla 84: Prueba de integración INT-07.....	102
Tabla 85: Prueba de integración INT-08.....	103
Tabla 86: Prueba de usabilidad y accesibilidad USL-01.....	103
Tabla 87: Prueba de usabilidad y accesibilidad USL-02.....	103
Tabla 88: Prueba de usabilidad y accesibilidad USL-03.....	104
Tabla 89: Prueba de usabilidad y accesibilidad USL-04.....	104
Tabla 90: Resultados de pruebas	105
Tabla 91: Métricas de la calidad del código	108
Tabla 92: Planificación de horas y días.....	113
Tabla 93: Costes de personal	114
Tabla 94: Costes del material	116

Tabla 95: Costes indirectos 116

Tabla 96: Coste Total del Proyecto..... 117

1 INTRODUCCIÓN

En la siguiente sección, se exponen las principales razones de elección y realización de dicho trabajo fin de grado. Así mismo, se especificarán los principales objetivos y la estructura que se seguirá a lo largo del documento con el fin de dar una idea general sobre el contenido de la memoria, antes de realizar una explicación en mayor profundidad sobre el proyecto desarrollado.

1.1 MOTIVACIÓN

Las razones que motivaron e influyeron en llevar a cabo dicho proyecto fue, principalmente, tener la oportunidad de aprender a desarrollar una aplicación web propia, en base a lo aprendido durante la formación recibida a partir de mis estudios. Por otro lado, existen diversas razones por las que decidí llevar a cabo dicha implementación basada en el sector turístico, las cuales son justificadas mediante un resumen histórico donde se mencionan los principales acontecimientos que incentivaron su creación y ciertas características a considerar.

A lo largo de la historia de la humanidad siempre ha existido un elemento social de querer escapar temporalmente de todo, dejando el ambiente cotidiano como principal motivo, sin preocuparse demasiado por el sitio al cual ir. Preferiblemente siempre se opta por un ambiente más agradable del que envuelve la rutina diaria, por tanto, en el caso del turismo este motivo constituye la base para el deseo de viajar e incluye la generación de una necesidad.

Desde el punto de vista académico, esta necesidad de viajar es abordada desde diferentes disciplinas, de manera que han sido desarrolladas diversas teorías y modelos en relación a este propósito. En el año 1943 Abraham Maslow, el cual era un psicólogo humanista, publicó un modelo de los elementos motivadores siguiendo una estructura jerárquica piramidal ordenada en cinco niveles [1]:

1. *Necesidades fisiológicas (en la parte inferior de la pirámide).*
2. *Necesidades de seguridad física y psíquica.*
3. *Necesidades de pertenencia social, afecto y relación.*
4. *Necesidades de estima y reconocimiento social.*
5. *Necesidades de autorrealización y desarrollo personal (en la parte superior).*



Ilustración 1: Modelo Piramidal de Maslow [1]

Esta teoría ha tenido más aplicaciones en el campo de estudio del sector turístico y entre otras se encuentra aquella elaborada por Pearce, el cual en base al modelo propuesto por Maslow propuso una adaptación incorporando el factor de la experiencia turística misma. De esta manera, los niveles existentes en el campo de la motivación turística se mostrarían ordenados de la siguiente manera (de menor a mayor importancia) [2]:

1. *Necesidad de relajación (descanso/actividad).*
2. *Necesidad de estimulación (seguridad/ emociones fuertes).*
3. *Necesidades sociales (de familia y relaciones íntimas de amistad).*
4. *Necesidades de autoestima (desarrollo personal, cultural, histórico, medioambiental).*
5. *Necesidades de autorrealización (búsqueda de la felicidad).*

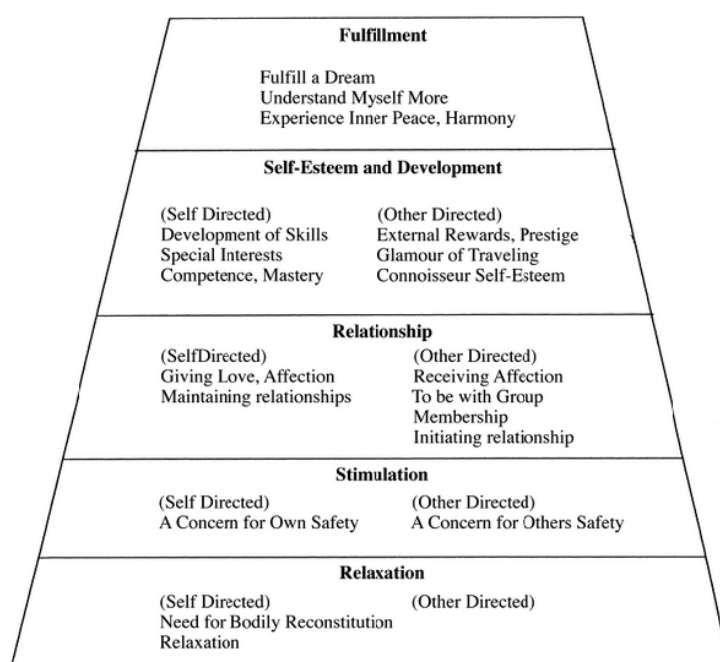


Ilustración 2: Modelo Piramidal de Pearce [2]

De este modo, es importante señalar que dicho modelo está basado en ciertos aspectos occidentales a diferencia de muchas partes del mundo donde el objetivo supremo sería el servicio a la comunidad en lugar de las necesidades de autorrealización individual.

Por consiguiente, el turismo ofrece una liberación de la represión que produce el trabajo y del resto de las obligaciones tomando distancia del entorno doméstico y además genera un deseo interior de querer adquirir nuevos conocimientos.

En referencia a los modelos piramidales de Pearce y Maslow, es preciso mencionar que estos fueron aquellos que han motivado a que la aplicación web se desarrolle en función de las necesidades de sus niveles superiores. De esta manera, dichos niveles se encuentran centrados en el deseo de querer aprender cosas nuevas, de experimentar diferentes culturas, de descubrirse a uno mismo, lo cual es parte de la búsqueda personal que el turista pretende encontrar en su destino.

En función de dichos aspectos mencionados, me pareció un área importante e imprescindible para hacer llegar la tecnología ya que como hemos observado se trata de una necesidad e incluso, como un derecho que contribuye, a su vez al desarrollo de los países y regiones.

1.2 OBJETIVOS

El objetivo principal del proyecto se basa en el diseño y la implementación de una aplicación web centrada en el sector turístico, como hemos mencionado anteriormente.

Dicha aplicación tratará de mostrar al usuario metadatos pertenecientes a una palabra (ya sea una persona, lugar, organización o evento, entre otros) introducida por el mismo en el buscador y mediante consultas a las bases de datos llamadas **DBpedia** y **Google Places**. Una vez obtenido los metadatos (los cuales son características asociadas a aquello que el usuario buscó), seleccionará uno o varios de ellos y eso permitirá la generación de uno o varios puntos de interés o end-points, mediante consultas realizadas a las bases de datos **DBpedia**, **Google Places** y **Foursquare**.

De este modo, se mostrarán en un mapa aquellos puntos de interés obtenidos por el usuario de forma que se generarán rutas en las que el propósito será completar dichos puntos de interés. Es necesario mencionar que la generación de rutas puede realizarse a partir de una o varias características provenientes de diferentes búsquedas realizadas por el usuario, es decir, la ruta podrá estar formada a partir de la búsqueda de diferentes características que no necesariamente tienen que tener algo en común.

Por consiguiente, se busca mejorar la satisfacción y experiencia del usuario acercando la tecnología en la visita de lugares distintos a su entorno habitual. Esto constituye una manera de entretenimiento para el cliente y a su vez permitirá afianzar los conocimientos acerca de personajes históricos, eventos, museos y hechos culturales destacados, entre otros, en relación a cada punto de interés.

De este modo, principalmente se busca crear una experiencia de usuario a través del mundo digital mediante el cumplimiento de rutas de manera que esto aportará la posibilidad, en cierto modo, de generar un ambiente lleno de retos y motivación a la hora de aprender la cultura general de cualquier región como si de un juego de tratase, pero en este caso amoldado a la vida real. También nos permite potenciar el turismo, lo cual también conllevará a obtener una cierta ganancia económica por parte de los lugares visitados gracias al usuario de la aplicación.

Además, es necesario comentar ciertas características que se deben cumplir:

- Su facilidad de uso.
- Debe ser capaz de proporcionar una información instantánea y ágil.
- Debe ser práctica y que ayude a organizarte.

1.3 ESTRUCTURA DEL DOCUMENTO

En el presente apartado se realiza una breve descripción de las diferentes partes en las que se encuentra estructurado el documento:

1. *Introducción*

Dicha sección se trata de aquella en la que nos encontramos. La misma constituye un apartado inicial cuyo propósito es el de contextualizar el texto fuente que está expuesto a continuación, dando una breve explicación o resumen de este.

2. Estado del arte

Como sabemos, el estado del arte es una modalidad de la investigación documental que permite el estudio del conocimiento acumulado (escrito en textos) dentro de un área específica. Por consiguiente, en este caso se llevará a cabo un análisis de aplicaciones web similares ya existentes en el mercado hasta el momento en relación con el sector turístico, además de la tecnología empleada para llevar a cabo el desarrollo del proyecto.

3. Análisis

En dicha sección se realizará una definición de aquello esperado por el sistema, donde se realizará una descripción general de la aplicación web y se especificará un listado de requisitos.

4. Diseño

En el presente apartado se determina de manera general como funcionará el sistema sin entrar en detalles incorporando consideraciones de la implementación tecnológica, como el hardware, la red, etc. Está compuesto por un diagrama de componentes del sistema que da respuesta a las funcionalidades descritas en la especificación de requisitos, un diagrama de paquetes, un diagrama de clases para mostrar las relaciones, operaciones y atributos de las clases principales que involucran al sistema y diversos diagramas de secuencia en los cuál se mostrará la interacción entre clases, componentes, subsistemas o actores con el sistema, en este caso más centrado en la interacción para el cálculo de rutas.

5. Pruebas realizadas

Sección en la que se especificará las pruebas realizadas acorde con los requisitos especificados en el apartado 3. *Análisis*.

6. Planificación y Presupuesto

Sección en la que se detallará la planificación del proyecto contemplando todas las tareas, incluyendo las tareas críticas, la secuenciación y la temporización. Además, se detallará el presupuesto del mismo.

7. Ética y Marco legal.

Sección en la que se menciona si existe alguna regulación, normativa técnica o legal o restricción que aplique al problema.

8. *Entorno socio-económico.* Aparecen referencias y comentarios justificados respecto al marco socio-económico en que se ubica el proyecto.

9. Conclusiones y Trabajo Futuro.

Sección en la que se exponen las conclusiones finales extraídas del proyecto y así como la verificación del debido cumplimiento de los objetivos descritos en los inicios del mismo. Además, se mencionaran las posibles mejoras o aplicaciones futuras como parte de la ampliación del presente proyecto.

10. Bibliografía.

11. Glosario: Acrónimos, Definiciones y Abreviaturas.

2 ESTADO DEL ARTE

El turismo y la tecnología van de la mano, el desarrollo continuo y sobre todo veloz de nuevas tecnologías, ayudan a la actividad turística a ofrecer nuevas experiencias y nuevas herramientas para el mejor desarrollo del turismo.

En un mundo sumamente comunicado e interrelacionado como lo es hoy, la manera en que las empresas o gobiernos puedan ir sumando estas herramientas tecnológicas y aplicarlas a la actividad turística, marcarán diferencias en cuanto a calidad de servicios y sobre todo la manera en que puedan darse a conocer en el mundo con los mismos.

La industria turística se está enfrentando actualmente a un nuevo entorno derivado de las transformaciones a las que se ha visto sometida. Este aspecto ha venido a modificar las tendencias del sector antes las demandas de un mercado que requiere concepciones más personalizadas y flexibles del producto o servicio turístico. Ante esta situación, las empresas necesitan orientar y centrar su modelo de gestión de negocio en las necesidades de sus clientes para poder afrontar los cambios y los retos estratégicos que tienen planteados. En este contexto, precisan más que nunca disponer de información de calidad sobre la actividad que desarrollan, el mercado en que actúan y los agentes con los que se interrelacionan.

Los aspectos señalados, convierten a la información y el conocimiento acerca del cliente en un activo clave para la gestión de la actividad turística que puede, además, ser determinante para mejorar el nivel de competitividad de las empresas de este sector. Sin duda, resulta fundamental que se doten de los instrumentos de Sistemas/Tecnologías de información más idóneos para obtener información valiosa que sirva de apoyo para mejorar la gestión de la actividad de negocio y la calidad del servicio al cliente.

Las empresas turísticas se están viendo sometidas actualmente a cambios rápidos que afectan a su dinámica de negocio y que han venido a configurar un nuevo entorno de negocio. En consecuencia, éstas necesitan disponer de información valiosa para mejorar la gestión del servicio, tomar decisiones con menor nivel de riesgo y desarrollar estrategias para obtener ventajas frente a la competencia. En este contexto, todas las estrategias competitivas que siguen las empresas dependen estrechamente de la información, que se convierte así en un recurso vital para esta industria.

De este modo, la propia esencia de la actividad turística caracterizada por una amplia diversidad de productos y destinos de gran complejidad, la heterogeneidad de los turistas y la sofisticación de sus demandas, convierte a la información en un factor crítico para estas empresas, dado que la organización de fuentes de información sobre cierto producto o destino turístico es fundamental para poder generar demanda y asegurar la satisfacción y fidelización del turista.

La importancia de la información en la industria turística se justifica también por las características propias que posee el producto turístico, que lo configura como un producto intensivo en información. Así, destacan la heterogeneidad del producto, compuesto por diversos componentes que intervienen en su definición, lo cual exige contar con conexiones continuas entre los diferentes agentes que intervienen en su producción para obtener una información actualizada, eficiente y exacta; su intangibilidad, derivada del hecho de que los consumidores potenciales de productos turísticos no puedan probarlo antes de su adquisición, obliga a las

empresas a dotar de mayor contenido informativo al producto y proveer información adicional al turista; su carácter perecedero, al tratarse de un producto que no es almacenable, precisa información que facilite el análisis de los problemas, situaciones y acciones para reducir los riesgos y la incertidumbre en las decisiones que se tomen; y, por último, el contexto internacional en que actúa la empresa turística hace que genere grandes volúmenes de información derivados de la necesidad de mantener comunicaciones continuas con los distintos agentes que participan en la cadena de distribución del producto.

Ante esta situación, disponer de información precisa, oportuna, y relevante es la base para que la empresa pueda tener éxito en la difícil tarea de satisfacer la demanda turística.

2.1 APLICACIONES SIMILARES

En base a lo comentado, a continuación se describen ciertas aplicaciones del sector turístico desarrolladas y puestas a disposición de los usuarios con el fin de realizar un análisis comparativo en relación con la aplicación web a desarrollar.

2.1.1 TripAdvisor

Se trata de una de las plataformas de recomendación del sector turístico más visitada, la cual fue fundada con el objetivo de crear una referencia de viajes en la web. De este modo, se trata de una aplicación encargada de planificar viajes a partir de las opiniones y críticas de sus usuarios junto con una amplia variedad de servicios de búsqueda y reserva.

Se pueden encontrar sitios donde ir a comer, donde dormir o lugares de interés turístico y de ocio alrededor de un destino turístico navegando a través de un mapa.

Funcionalidades principales [3]:

- Permite a los usuarios leer las opiniones acerca de personas que ya visitaron el lugar de destino seleccionado.
- Nos muestra información sobre cualquier destino buscado.
- Posibilita el acceso a la compra de billetes de avión, reservas de hoteles o alquileres vacacionales.
- Permite añadir tus propias opiniones, videos y fotos sobre cualquier destino.
- Los viajeros pueden ver en el mapa la puntuación y opiniones de otros viajeros para comparar varias opciones.

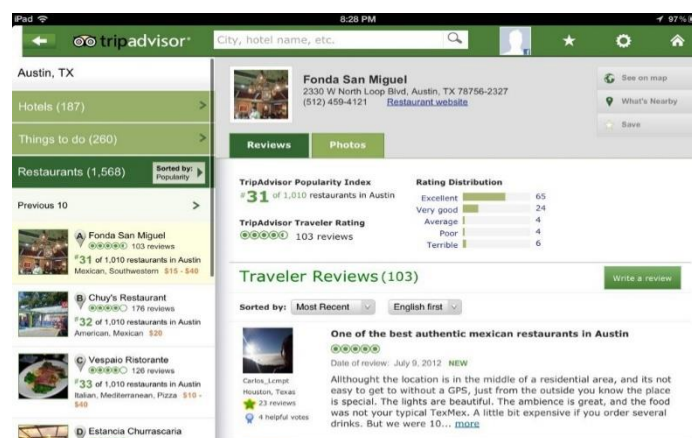


Ilustración 3: Aplicación web TripAdvisor [3]



Ilustración 4: Planificación de viaje en TripAdvisor [3]

2.1.2 TouristEye

Se trata de una aplicación web que está orientada a la planificación de un viaje añadiendo diferentes lugares de interés durante el recorrido, a modo lista de deseos o pendientes de visitar. Esta aplicación tiene funciones gratuitas, aunque algunas son de pago. La comunidad añade cada día nuevos puntos de interés geolocalizados, por tanto, hay una buena cantidad de lugares para visitar en las ciudades que tengamos añadidas a nuestro viaje: sitios de interés turístico, restaurantes y hoteles.

Funcionalidades principales [4]:

- Colecciona ideas para tus viajes y escapadas.
- Recomendación de sitios alrededor de tu destino de interés.
- Cada destino dispone de fotos, comentarios y su ubicación en mapa.
- Se muestra la valoración de los destinos de viaje realizada por otros viajeros que ya lo visitaron.
- Permite a los usuarios crear sus propias rutas.



Ilustración 5: Aplicación web TouristEye [4]



Ilustración 6: Agregación de planes con TouristEye [4]

2.1.3 Minube.

Minube consiste en una plataforma para organizar viajes y relatar después las experiencias, siendo una red social destinada a viajeros y aventureros. Así pues, puede ser utilizada de dos maneras diferentes: como una herramienta para buscar vuelos y alojamiento en una ciudad a la par que descubres sitios, o tan solo como una red social en la que conocer una extensa comunidad de viajeros dispuestos a compartir sus experiencias con los demás.

Funcionalidades principales [5]:

- Si no conoces el destino aún donde viajar, te permite inspirarte mediante listas temáticas de lugares.
- Posibilidad de poder crear tu propia guía de viaje del destino que desees en la cual incluirá:
 - Lugares qué ver.
 - Establecimientos dónde comer.
 - Sitios dónde poder hospedarse.
- Proporciona la funcionalidad de añadir fotos, multimedia y archivos a tu perfil de usuario.
- Muestra puntos de interés alrededor de tu destino en el mapa.

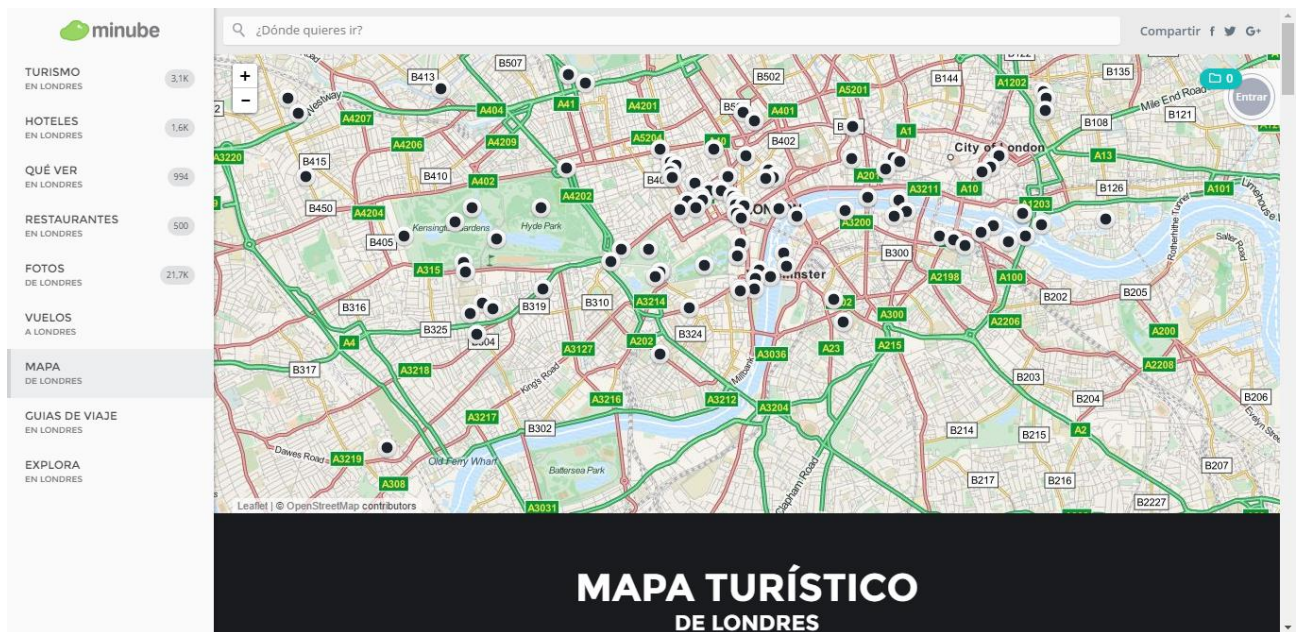


Ilustración 7: Aplicación web minube [5]

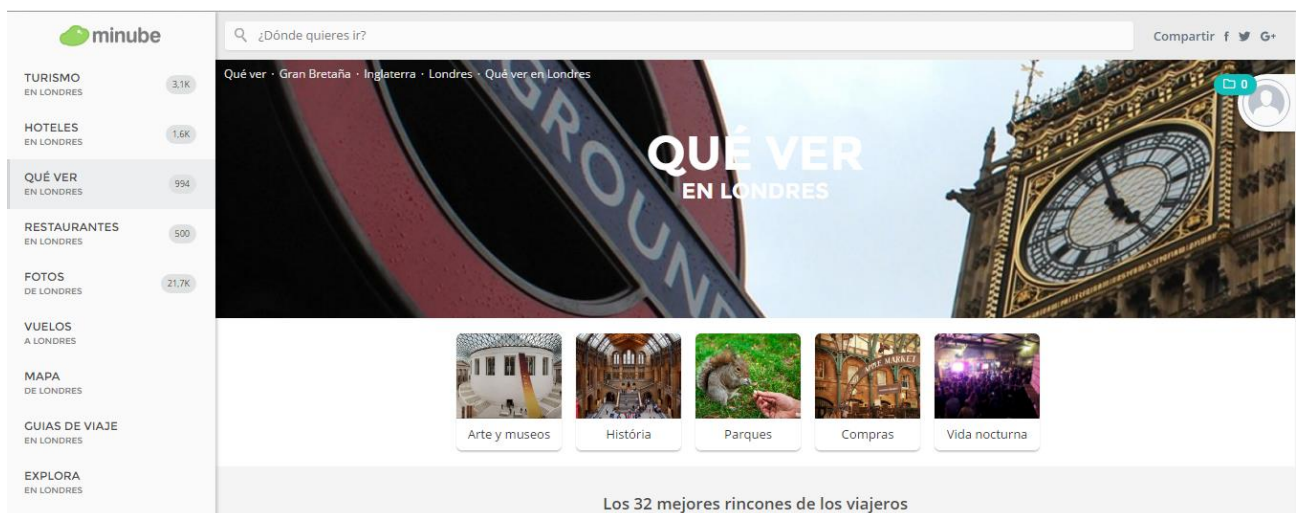


Ilustración 8: Puntos de interés en el destino con minube [5]

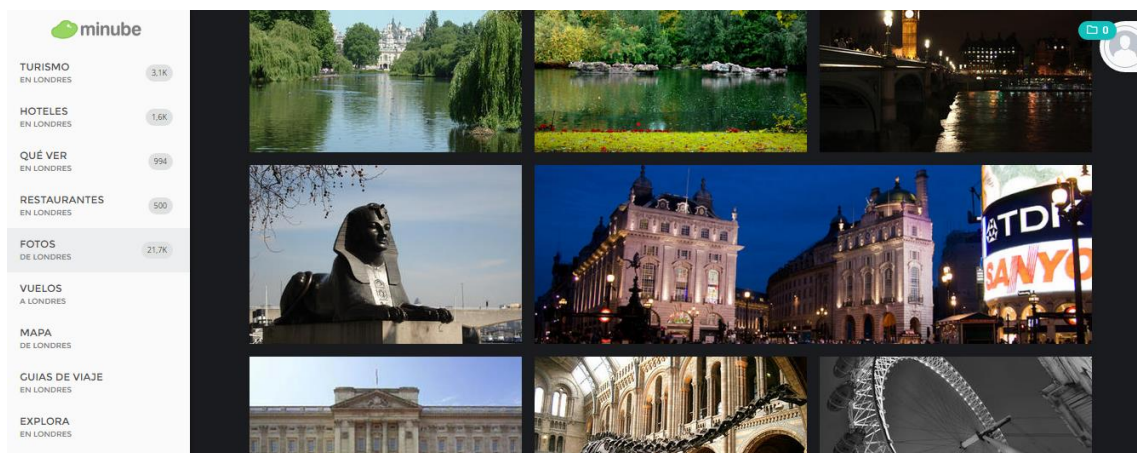


Ilustración 9: Fotos del lugar destino con minube [5]

2.1.4 Triplt

Triplt se trata de una aplicación web cuyo objetivo principal es el de planificar viajes, al igual que aquellas descritas anteriormente. Por consiguiente, proporciona la búsqueda de un determinado destino mostrando horarios, check-ins en hoteles, tiempo previsto, mapas asociados y otros datos de interés a la hora de realizar el viaje.

Funcionalidades principales [6]:

- Permite al usuario realizar una planificación personalizada de su viaje añadiendo diferentes actividades asociadas a cada día que el usuario se encuentre en su destino.
- Permite añadir o eliminar planes manualmente.
- Sincroniza los planes de viaje con tu calendario.
- Posibilita compartir rutas turísticas realizadas y otros datos entre los distintos amigos que se tengan en Triplt.

The screenshot shows the 'Add a Trip' form on the Triplt website. At the top, there's a navigation bar with the Triplt logo and links for Home, Trips, Network, Point Tracker, Triplt Pro, and Teams. A banner at the top right says 'Triplt Pro just saved \$144 on airfare from CHS to JFK' and 'Upgrade to Triplt Pro now!'. Below the navigation bar, there's a 'Share your trips' button. The main form is titled 'Add a Trip' and includes a sub-header: 'Add a trip by entering your destination city and trip dates. Or just forward your confirmation emails to plans@tripit.com, and we'll create the trip for you.'

The form fields include:

- Destination City***: A text input field containing 'London, United Kingdom'.
- Trip Name**: A text input field containing 'Your trip'.
- Trip image**: A button with an airplane icon and a 'Change Image' link below it.
- Start Date***: A date picker showing '06/05/2016'.
- End Date***: A date picker showing '06/09/2016'.
- Private**: A checkbox that is unchecked.
- Business**: A checkbox that is checked.
- Trip description**: A text area for describing the trip.
- I'm a traveler on this trip.**: A checkbox that is checked.

At the bottom of the form, there are 'Cancel' and 'Add Trip' buttons. To the right of the form, there's a sidebar with the heading 'An easier way to create trips' and the email address plans@tripit.com, with a note: 'Just forward your confirmation emails to this address and we'll condense it into one itinerary.'

Ilustración 10: Aplicación web Triplt [6]

The screenshot shows the Triplt trip planning interface for 'London, United Kingdom, June 2016'. The main header displays the trip details: 'Jun 5 - 9, 2016 / London, United Kingdom'. Below this, there are sections for 'Travelers' (Marta Plaz... Add), 'Viewers' (Add), and 'Planners' (Add). There are also icons for settings, a calendar, and a list of activities. A sidebar on the right shows social media sharing options (Email, Print, Facebook, Twitter, LinkedIn).

The main content area shows a calendar view for 'Jun 5 - 9'. Each day has a weather forecast and a 'ADD PLANS' button. The weather for each day is:

- Sun: PARTLY CLOUDY, 76°F / 54°F, + ADD PLANS
- Mon: PARTLY CLOUDY, 74°F / 53°F, + ADD PLANS
- Tue: RAIN, 73°F / 55°F, + ADD PLANS
- Wed: PARTLY CLOUDY, 70°F / 51°F, + ADD PLANS
- Thu: PARTLY CLOUDY, 70°F / 45°F, + ADD PLANS

An 'Add plans' modal window is open, showing a list of activities categorized into Transportation, Activities, and Other. The activities listed are:

- Transportation**: Flight, Car Rental, Rail, Cruise, Parking, Other (Ferry, etc.)
- Activities**: Lodging, Restaurant, Meeting, Activity (Theatre, Concert, etc.)
- Other**: Map, Directions, Note

At the bottom of the modal, there's a 'tripit.com. Or, [add plans](#).' link.

Ilustración 11: Planificación de actividades con Triplt [6]

2.1.5 The Culture Trip

Se trata de una aplicación web que te permite descubrir lo mejor de la comida, el arte, el cine, la literatura, viajes, entre otros aspectos culturales de cada país del mundo.

Puedes cultivar el ciudadano del mundo que cada usuario tiene dentro de sí mismo con el contenido cultural que contiene dicha aplicación proveniente de historias y recomendaciones para cada ciudad por parte de 15 sucursales asociadas.

Funcionalidades principales [7]:

- Permite la búsqueda de información cultural de una determinada localización.
- Las búsquedas realizadas se pueden realizar mediante una temática cultural:
 - Arte.
 - Comida.
 - Viajar.
 - Música.
 - Cine.
 - Literatura.
 - Diseño.
 - Historia.
 - Otros.

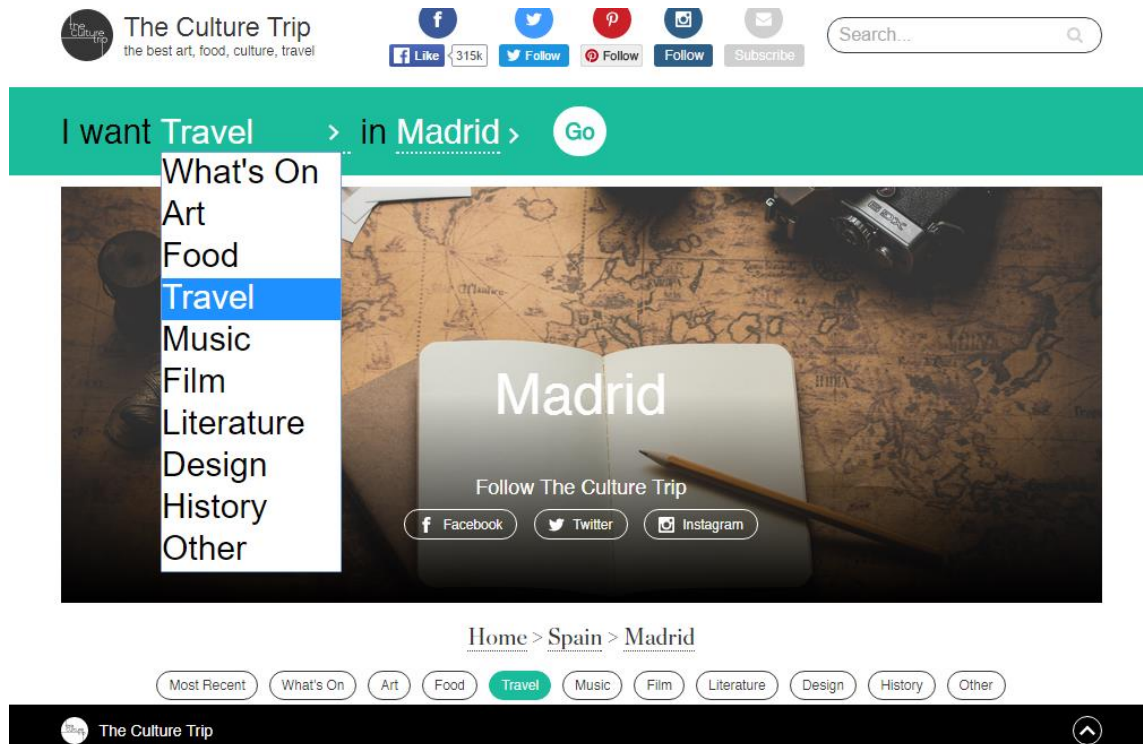


Ilustración 12: Aplicación web The Culture Trip [7]

2.2 ANÁLISIS COMPARATIVO

Tras el estudio de los diferentes aspectos sobre las aplicaciones similares a la aplicación web a desarrollar, en el presente apartado se procede a realizar un análisis comparativo de las mismas teniendo en cuenta aquellas funcionalidades más importantes con el fin de identificar aquellas que fueron consideradas relevantes para la implementación del proyecto. A continuación, se muestra una tabla en la que se especificarán dichas funcionalidades indicando si se cumplen (SI) o no se cumplen (NO) en relación a cada una de las aplicaciones web similares explicadas en el apartado anterior:

Funcionalidad	TripAdvisor	TouristEye	Minube	Triplt	The Culture Trip
Ubicación de localizaciones en el mapa	SI	SI	SI	SI	NO
Añadir/Eliminar POIs	NO	SI	SI	SI	NO
Mostrar información multimedia de cada POI	SI	SI	SI	NO	SI
Buscar destino mediante temática	NO	NO	SI	NO	SI
Generar ruta personalizada manualmente	NO	SI	SI	SI	NO
Generar ruta automáticamente	NO	NO	NO	NO	NO
Mostrar información cultural según localización	SI	SI	SI	NO	SI
Buscar por entidad	NO	NO	NO	NO	NO
Reserva de hoteles y vuelos	SI	NO	SI	SI	NO
Valoración de cada localización	SI	SI	SI	NO	NO
Valoración de ruta generada	NO	NO	NO	NO	NO
Recomendación de POIs alrededor del destino	SI	SI	SI	NO	SI

Tabla 1: Análisis comparativo de aplicaciones web similares

Como podemos observar en la tabla anterior, no todas las aplicaciones web cumplen todas las funcionalidades indicadas. De este modo, gracias a dicho análisis resultó la necesidad de añadir al proyecto ciertos aspectos en común con las mismas que fueron considerados útiles, pero por otro lado se descartaron otros que fueron irrelevantes en relación con el objetivo buscado.

Por consiguiente, a continuación se mencionarán aquellas funcionalidades que se han considerado más importantes para la confección del proyecto y, así como sus justificaciones:

- **Ubicación de localizaciones en el mapa:** Respecto a dicha funcionalidad, todas las aplicaciones excepto The Culture Trip la cumplen. Se consideró un aspecto importante ya que permite establecer una relación entre la tecnología y el mundo real permitiendo que el usuario asocie la lejanía o cercanía de cada punto de interés que desea visitar.
- **Buscar por entidad:** Con respecto a dicha funcionalidad, podemos afirmar que no ha sido implementada por ninguna de las aplicaciones similares de tal manera que se consideró una mejora tener la posibilidad de no sólo buscar por una localización, sino que se tuviera la posibilidad de buscar por una entidad (persona, lugar, organización o evento, entre otros) con el fin de ampliar el área de búsqueda para planificar el viaje deseado. De este modo, al buscar por entidad la aplicación web a desarrollar obtendrá como resultado un conjunto de características que serán seleccionadas por el usuario según su interés, de las cuales se extraerán localizaciones en el proceso de generación de rutas.
- **Añadir/Eliminar POIs:** En relación a dicho aspecto, las aplicaciones similares que lo cumplían podían añadir puntos de interés alrededor del destino buscado, sin embargo, se decidió realizar una mejora de dicha funcionalidad permitiendo no añadir o eliminar POIs que necesariamente se encontrarán en los alrededores de un destino buscado sino que pudieras añadir o eliminar aquellas características que estuvieran asociados a la entidad buscada (persona, lugar, organización o evento, entre otros) o podrían no tener ninguna relación. Es decir, en lugar de que los puntos de interés tuvieran relación por la distancia al destino buscado, se pretendía que los mismos tuvieran relación por la entidad buscada o sin necesariamente tener una relación entre los mismos.
- **Mostrar información multimedia de cada POI:** Dicha funcionalidad ha sido adoptada por la mayoría de las aplicaciones similares y se consideró de gran utilidad para que el usuario tuviera mayor conocimiento acerca de los lugares que va a visitar durante su viaje. Esta información consta de la imagen del lugar, descripción, fuente de búsqueda de la que proviene, nombre, dirección, entre otros.
- **Generar ruta automáticamente:** Dicha funcionalidad se trata de uno de los puntos clave de la aplicación web a desarrollar. Como podemos observar, en ninguna de las aplicaciones similares no se encuentra implementada ya que no todas pueden generar rutas como tal, solamente alguna de ellas puede generar rutas personalizadas manualmente incluyendo actividades o puntos de interés alrededor del destino buscado realizando una planificación de visitas durante el viaje, por tanto, se encontró una clara necesidad de generar rutas con una planificación automática. De este modo, en el proceso de generación automática de rutas, el usuario tendrá la única tarea de añadir o eliminar aquellas características asociadas a la entidad buscada y la aplicación web automáticamente obtendría una ruta con los puntos de interés asociados con las características seleccionadas por parte del usuario y ordenados con el criterio de la distancia más corta entre los mismos, sin necesidad de que sea el usuario el que organice y planifique el orden de las visitas realizadas.

- **Valoración de ruta generada:** Tal y como hemos podido observar en la mayoría de las aplicaciones analizadas se da la posibilidad de realizar una valoración sobre un lugar visitado, sin embargo, en el caso de las aplicaciones que tienen la posibilidad de generar rutas de manera manual tampoco se tiene la posibilidad de valorar la ruta completa. De este modo, una mejora incluida en el proyecto fue la posibilidad de valorar las rutas generadas.

3 ANÁLISIS

En el presente apartado, se detallará aquello que se espera del sistema lo cual permitirá identificar aquellas necesidades del producto software a desarrollar, es decir, las funcionalidades requeridas de la aplicación web que constituirá el proyecto. Para ello, en primer lugar se especificarán los requisitos de usuario realizados en paralelo a los diagramas de casos de uso, los cuales son empleados con el fin de especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. A continuación, a partir de los mismos se extraerán los requisitos de software.

3.1 REQUISITOS DE USUARIO.

Es común ver que existen muchas diferencias entre el cliente y el desarrollador ya que muchas veces se les dificulta el satisfacerlo y no pueden cumplir con todos sus requisitos impuestos, dando lugar a que en algunos casos se cumple con lo que pide o solicita el cliente pero en otras ocasiones no se puede cumplir con lo esperado. Por tanto, se observa una clara necesidad de tener en cuenta al usuario durante todo el proceso de desarrollo del proyecto y para ello ha sido necesario realizar la especificación de los requisitos de usuario, con el fin de determinar los requisitos abstractos de alto nivel.

Por consiguiente, los requisitos de usuario describen en un lenguaje natural los requerimientos funcionales y no funcionales de tal forma que sean comprensibles por los usuarios del sistema que no posean un conocimiento técnico detallado. Únicamente se especificará el comportamiento externo del sistema y se evitará, tanto como sea posible, las características de diseño del sistema.

A continuación, se muestra la plantilla empleada para determinar los requisitos de usuario:

Identificador: RU-XX	
Título	
Fuente	
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	

Tabla 2: Plantilla definición requisitos de usuario

Los campos recogidos en la anterior tabla tiene el siguiente significado:

- **Identificador:** Permite identificar unívocamente a cada requisito en función de la codificación RU-XX:
 - RU: Se refiere al tipo de requisito. En este caso, al tratarse de requisitos de usuario, siempre tomará el valor RU.
 - XX: Hace referencia al número identificador del requisito.
- **Título:** Título asignado al requisito.
- **Fuente:** Determina de qué persona o conjunto de personas proviene el requisito.
- **Prioridad:** Especifica con qué relevancia considera el cliente que se tiene que tener en cuenta dicho requisito durante el proceso de implementación.

- **Descripción:** Se realiza una explicación acerca de la funcionalidad o restricción que el requisito va a implicar en el proyecto.

De este modo, se procede a determinar los requisitos de usuario:

Identificador: RU-01	
Título	Búsquedas
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá realizar búsquedas de cualquier tipo de entidad.

Tabla 3: Búsquedas RU-01

Identificador: RU-02	
Título	Tipo de Búsquedas
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La entidad buscada podrá ser una persona, localización, organización, evento u otros.

Tabla 4: Tipo de Búsquedas RU-02

Identificador: RU-03	
Título	Caracterización entidad
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá visualizar características en relación a la entidad buscada.

Tabla 5: Caracterización entidad RU-03

Identificador: RU-04	
Título	Características seleccionadas/descartadas
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá seleccionar/descartar las características disponibles según el tipo de entidad.

Tabla 6: Características seleccionadas/descartadas RU-4

Identificador: RU-05	
Título	Generación ruta
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá generar rutas a partir de las características que seleccionó.

Tabla 7: Generación ruta RU-05

Identificador: RU-06	
Título	Localizaciones en el mapa
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá observar en un mapa las localizaciones asociadas a las características que seleccionó.

Tabla 8: Localizaciones en el mapa RU-06

Identificador: RU-07	
Título	Punto de Origen
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario deberá indicar el punto de origen donde desea iniciar la ruta.

Tabla 9: Punto de Origen RU-07

Identificador: RU-08	
Título	Completar ruta
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá visualizar el orden en el que debe visitar los puntos de interés que constituyen la ruta generada.

Tabla 10: Completar ruta RU-08

Identificador: RU-09	
Título	Valoración ruta
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá valorar cada ruta generada.

Tabla 11: Valoración ruta RU-09

Identificador: RU-10	
Título	Información por localización
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá hacer click sobre cada localización en el mapa y podrá visualizar información en relación al mismo.

Tabla 12: Información por localización RU-10

Identificador: RU-11	
Título	Idioma de las características
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El sistema podrá mostrar las características obtenidas tras una búsqueda en Castellano o en Inglés.

Tabla 13: Idioma de las características RU-11

Identificador: RU-12	
Título	Idioma Interfaz
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El sistema mostrará la interfaz en Castellano.

Tabla 14: Idioma Interfaz RU-12

Identificador: RU-13	
Título	Interfaz adaptativa
Fuente	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación web será adaptativa a cualquier dispositivo que la use.

Tabla 15: Interfaz adaptativa RU-13

3.2 CASOS DE USO.

Tal y como hemos comentado anteriormente, los casos de uso mostrarán la secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

De este modo, a continuación se mostrará un diagrama de casos de uso, donde se especificarán los actores y una serie de tablas explicativas en la que se definirán cada uno de los casos de uso mostrados en el diagrama.

3.2.1 Diagrama de casos de uso.

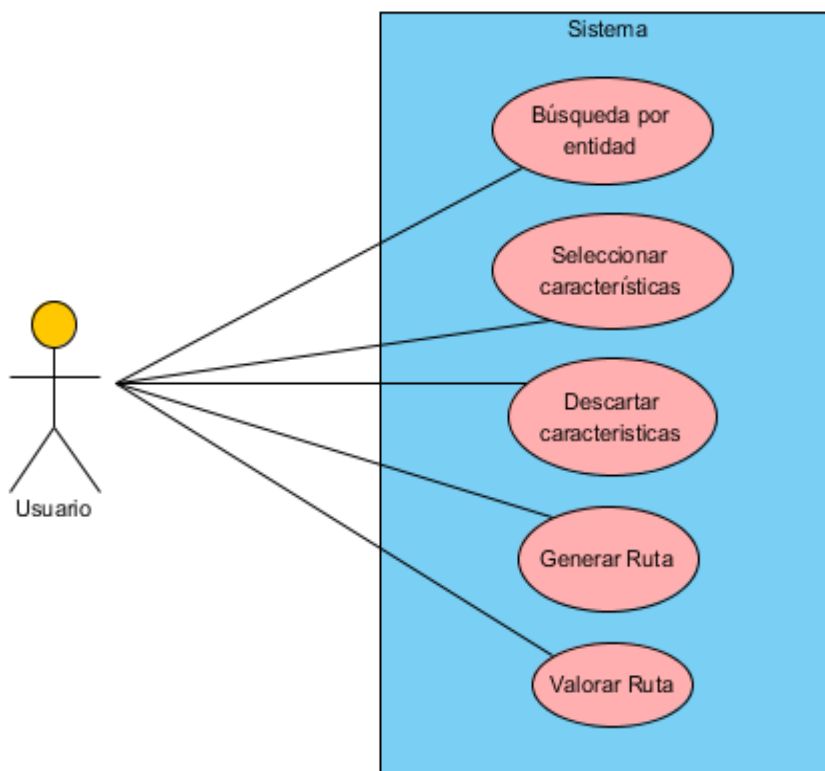


Ilustración 13: Diagrama de casos de uso

3.2.2 Actores

Como sabemos, se llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también a todos los sistemas externos, además de entidades abstractas, como el tiempo.

En este caso, se ha considerado un único rol para llevar a cabo la interacción con el sistema:

- **Usuario:** Se trata del tipo de rol asociado al operador humano que actúa sobre la aplicación web. Este actor podrá ser cualquier tipo de persona que haga uso de la aplicación sin que sea necesario que el mismo tenga un conocimiento previo acerca del uso del sistema. Para que el mismo pueda interactuar con el sistema, será necesario que disponga de cualquier dispositivo con conexión a Internet.

3.2.3 Descripción de los casos de uso

Para la especificación de cada caso de uso, se ha empleado la siguiente plantilla:

Identificador: CU-XX	
Título	
Actor	
Objetivo	
Precondiciones	
Postcondiciones	
Escenario básico	

Tabla 16: Plantilla definición casos de uso

Donde:

- **Identificador:** Permite identificar unívocamente a cada requisito en función de la codificación CU-XX:
 - CU: Se refiere al tipo de aspecto a tratar. En este caso, al tratarse de casos de uso, siempre tomará el valor CU.
 - XX: Hace referencia al número identificador del requisito.
- **Título:** Nombre asignado al caso de uso.
- **Actor:** Qué entidad externa al sistema interactúa con la aplicación web.
- **Objetivo:** Caso que cubre.
- **Precondiciones:** Condiciones necesarias antes de realizar la acción descrita.
- **Postcondiciones:** Condiciones que se tienen que cumplir una vez realizada la acción.
- **Escenario básico:** Descripción del caso de uso.

Identificador: CU-01	
Título	Búsqueda por entidad.
Actor	Usuario
Objetivo	Obtener un conjunto de características en relación a la entidad buscada.
Precondiciones	Introducir la entidad deseada en el buscador.
Postcondiciones	Mostrar características en relación a la entidad buscada.
Escenario básico	<ul style="list-style-type: none"> -Acceder a la aplicación web (en caso de que el usuario no haya accedido antes). -Introducir la entidad deseada en el buscador. -Mostrar características obtenidas.

Tabla 17: CU-01 Búsqueda por entidad

Identificador: CU-02	
Título	Seleccionar características
Actor	Usuario
Objetivo	Seleccionar características relacionadas con la entidad buscada.
Precondiciones	<ul style="list-style-type: none"> -El usuario deberá haber realizado una o varias búsquedas previamente. -El usuario debe seleccionar una o varias características de la sección "Resultados Obtenidos".
Postcondiciones	<ul style="list-style-type: none"> -El usuario podrá observar las características seleccionadas en la sección "Aspectos Seleccionados". -Las características que fueron seleccionadas no deberán de mostrarse en la sección "Resultados Obtenidos".
Escenario básico	<ul style="list-style-type: none"> -Acceder a la aplicación web (en caso de que el usuario no haya accedido antes). -Introducir la entidad deseada en el buscador (si antes no se ha realizado ninguna búsqueda). -Seleccionar las características deseadas.

Tabla 18: CU-02 Seleccionar características

Identificador: CU-03	
Título	Descartar características
Actor	Usuario
Objetivo	Descartar características que fueron seleccionadas.
Precondiciones	<ul style="list-style-type: none"> -El usuario deberá haber realizado una o varias búsquedas previamente. -El usuario debe descartar una o varias características de la sección "Aspectos Seleccionados".
Postcondiciones	<ul style="list-style-type: none"> -El usuario podrá observar las características descartadas en la sección "Resultados Obtenidos". -Las características que fueron descartadas no deberán de mostrarse en la sección "Aspectos Seleccionados".
Escenario básico	<ul style="list-style-type: none"> -Acceder a la aplicación web (en caso de que el usuario no haya accedido antes). -Introducir la entidad deseada en el buscador (si antes no se ha realizado ninguna búsqueda). -Seleccionar las características deseadas. -Descartar características del conjunto seleccionado.

Tabla 19: CU-03 Descartar características

Identificador: CU-04	
Título	Generar ruta.
Actor	Usuario
Objetivo	Generar ruta a través de las características seleccionadas por el usuario.
Precondiciones	<ul style="list-style-type: none"> -El usuario debe haber seleccionado dos o más características. -El usuario debe indicar el punto de origen donde desea iniciar la ruta.
Postcondiciones	Mostrar localizaciones asociadas a cada característica en el orden que deben de visitarse empezando por el punto de origen indicado por el usuario, lo cual constituye la ruta.
Escenario básico	<ul style="list-style-type: none"> -Acceder a la aplicación web (en caso de que el usuario no haya accedido antes). -Introducir la entidad deseada en el buscador (si antes no se ha realizado ninguna búsqueda). -Seleccionar las características deseadas. -Pulsar sobre la opción “Generar ruta”.

Tabla 20: CU-04 Generar ruta

Identificador: CU-05	
Título	Valorar ruta.
Actor	Usuario
Objetivo	Puntuar una ruta generada previamente.
Precondiciones	Generar una ruta.
Postcondiciones	Almacenar la valoración realizada por el usuario.
Escenario básico	<ul style="list-style-type: none"> -Acceder a la aplicación web (en caso de que el usuario no haya accedido antes). -Introducir la entidad deseada en el buscador (si antes no se ha realizado ninguna búsqueda). -Seleccionar las características deseadas. -Pulsar sobre la opción “Generar ruta”. -Realizar la calificación de la ruta y pulsar el botón “Valorar”.

Tabla 21: CU-05 Valorar ruta

3.3 REQUISITOS DE SOFTWARE

Tras la realización de los casos de uso, se procede a especificar los requisitos necesarios para las posteriores fases del desarrollo del proyecto. Dichos requisitos constituirán la base fundamental para el funcionamiento de la aplicación web.

De este modo, es necesario mencionar que el proceso de análisis de requisitos exige continuas realimentaciones, por lo que el catálogo de requisitos no será nunca definitivo, pues a medida que se profundice en el análisis surgirán nuevos requisitos que no se tuvieron en cuenta, o desaparecerán requisitos que estaban mal planteados.

Es necesario mencionar, que a continuación se detallará la Especificación de Requisitos Software (ERS) según las directrices dadas mediante el estándar ANSI/IEEE 830, 1998.

3.3.1 Propósito

El propósito de esta sección consistirá en definir las especificaciones tanto funcionales como no funcionales para el desarrollo de una aplicación web basada en la planificación inteligente y contextualizada de rutas de turismo.

3.3.2 Alcance

En referencia al alcance del sistema, es necesario mencionar que el proyecto estará constituido por una aplicación web que tendrá asignado el nombre de “*In-Tour Web*” con el fin de realizar una planificación inteligente y contextualizada de rutas turísticas.

Dicha aplicación web estará comprometida con diferentes Stakeholders cuyos intereses pueden ser contradictorios. Para poder tener en cuenta a cada uno de ellos en el desarrollo del proyecto, se definen a continuación junto con sus respectivos intereses:

- **Desarrolladores:** Serán los usuarios que implementarán el producto a desarrollar, basándose en otras aplicaciones similares y en los intereses del cliente para implementar la aplicación web. Sus intereses son un diseño claro y lo más sencillo posible en el que basarse y la modularidad del mismo, para poder dividir la carga de trabajo de forma eficaz. En este caso, este tipo de usuarios estará constituido por un miembro, el cual es el autor del presente documento.
- **Cliente:** Se trata de la empresa o persona que ha llegado a un acuerdo con los desarrolladores con el fin de construir la aplicación web. Sus intereses son: el precio del diseño, pero también lo que le va a suponer el desarrollo de la aplicación *In-Tour Web*, que cumpla todas las funcionalidades pedidas y que el producto final sea de fácil mantenimiento. En este caso, el cliente será el tutor que ha dirigido al desarrollador a lo largo del proyecto.
- **Usuarios finales:** Se trata de cada una de las personas que utilizará *In-Tour Web*. Sus intereses serán la velocidad a la que se ejecuta la aplicación, los recursos que consumirá en el dispositivo que empleen, la disponibilidad y la facilidad de uso. En el caso de dicho proyecto, estará dirigido aquellas personas que necesiten ayuda a la hora de realizar la planificación de su viaje de una manera dinámica y contextualizada.

3.3.3 Personal involucrado

NOMBRE	JOSÉ MARÍA ÁLVAREZ RODRÍGUEZ
ROL	Cliente
CATEGORÍA PROFESIONAL	Tutor
RESPONSABILIDAD	Dirigir y guiar el proyecto a realizar
INFORMACIÓN DE CONTACTO	joalvare@inf.uc3m.es

Tabla 22: Ficha descriptiva José María Álvarez Rodríguez

NOMBRE	MARTA PLAZA RODRÍGUEZ
ROL	Analista, diseñador y programador
CATEGORÍA PROFESIONAL	Alumno
RESPONSABILIDAD	Análisis y diseño del proyecto al igual que la implementación del mismo.
INFORMACIÓN DE CONTACTO	100303516@alumnos.uc3m.es

Tabla 23: Ficha descriptiva Marta Plaza Rodríguez

3.3.4 Descripción General

En la presente sección se llevará a cabo una descripción de aquellos factores que afectan al producto y a sus requisitos. No se van a describir los requisitos como tal, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3.3.5. *Especificación de Requisitos*, haciendo que sean más fáciles de entender.

3.3.4.1 Perspectiva del producto

En referencia a la perspectiva del producto, a continuación se mencionarán aquellos aspectos que van a describir el producto en su entorno: tipo de clientes para el que es dirigido, área de aplicación, beneficios obtenidos mediante su uso...etc. Además, se mencionará la dependencia/independencia con otros sistemas así como se indicarán las justificaciones de por qué hacer uso del mismo frente a los productos ofrecidos por sus consumidores.

- **Tipo de clientes al que se dirige:** El sistema a desarrollar está dirigido a usuarios que cumplan las siguientes características:
 - Edad: De 14 a 80 años.
 - Personas que tengan las posibilidades económicas para viajar.
 - Personas indecisas en cuanto a la planificación de su viaje.
 - Personas amantes de la cultura y el turismo.
- **Área de aplicación:** Sector turístico.
- **Necesidad cubierta:**
 - Datos de interés acerca de personajes históricos, aspectos culturales, monumentos, organizaciones, eventos...etc.
 - Planificación de viajes.
 - Control sobre el orden en el que visitar los puntos de interés que constituyen la ruta más corta.

- **Razones por las que hacer uso del mismo:** Permitirá a sus usuarios organizar viajes de manera automática sin necesidad de calcular la ruta más corta entre los lugares que desea visitar y adquirir conocimientos culturales, históricos y de actualidad en relación a cada punto de interés visitado.
- **Principales competidores:** Como se mencionó en el apartado 2. *Estado del Arte* no existen competidores en concreto para este producto, pero si aplicaciones web similares de las cuales se han tenido en cuenta las características más beneficiosas de las mismas.
- **Principales aspectos diferenciadores:** Frente a las aplicaciones web similares, se dan dos aspectos principales que hace que este producto software se diferencie del resto:
 - Generación de la ruta más corta de manera automática.
 - Buscar localizaciones a partir de cualquier tipo de entidad: persona, localización, organización, evento, entre otros.
- **Dependencia/Independencia con otros sistemas:** El producto no es parte de ningún sistema mayor pero sí tiene una relación de dependencia con las bases de datos DBpedia, Google Places y Foursquare.

3.3.4.2 Funcionalidad del producto

A continuación se mencionarán las principales funcionalidades que proporcionará el producto software:

- **Búsqueda por entidad:** El sistema permitirá no sólo buscar por una localización, sino que el usuario tendrá la posibilidad de buscar por una entidad (persona, lugar, organización o evento, entre otros) con el fin de ampliar el área de búsqueda para planificar el viaje deseado. Como resultado de la búsqueda se obtendrá un conjunto de características que serán seleccionadas por el usuario según su interés.
- **Añadir/Eliminar Características:** El sistema permitirá que el usuario añada o elimine aquellas características que estuvieran asociados a la entidad buscada (persona, lugar, organización o evento, entre otros) o podrían no tener ninguna relación con otras características obtenidas a partir de búsquedas anteriores.
- **Generación de rutas automáticamente:** Dicha funcionalidad se basa en generar rutas a partir de una planificación automática. De este modo, el usuario tendrá la única tarea de añadir o eliminar aquellas características deseadas y la aplicación web automáticamente obtendría una ruta con los puntos de interés asociados con las características seleccionadas por parte del usuario y ordenados con el criterio de la distancia más corta entre los mismos, sin necesidad de que sea el usuario el que organice y planifique el orden de las visitas realizadas.
- **Ubicación de localizaciones en el mapa:** El sistema mostrará en el mapa aquellos puntos de interés o POIs asociados cada uno de ellos a cada una de las características seleccionadas por parte del usuario para generar la ruta.
- **Mostrar información multimedia de cada POI:** El sistema permitirá que el usuario haga click sobre cualquier localización marcada en el mapa y se mostrará la siguiente información: la imagen del lugar, descripción, fuente de búsqueda de la que proviene, nombre y dirección, entre otros.
- **Valoración de ruta generada:** El sistema permitirá valorar las rutas generadas sin necesidad de tener que valorar cada punto que constituya la ruta.

Como se mencionó anteriormente el producto es dependiente de otros sistemas que le permitirán realizar la mayoría de las funciones mencionadas. Por tanto, a continuación se muestra un diagrama en el que se podrán visualizar las distintas relaciones con dichos sistemas:

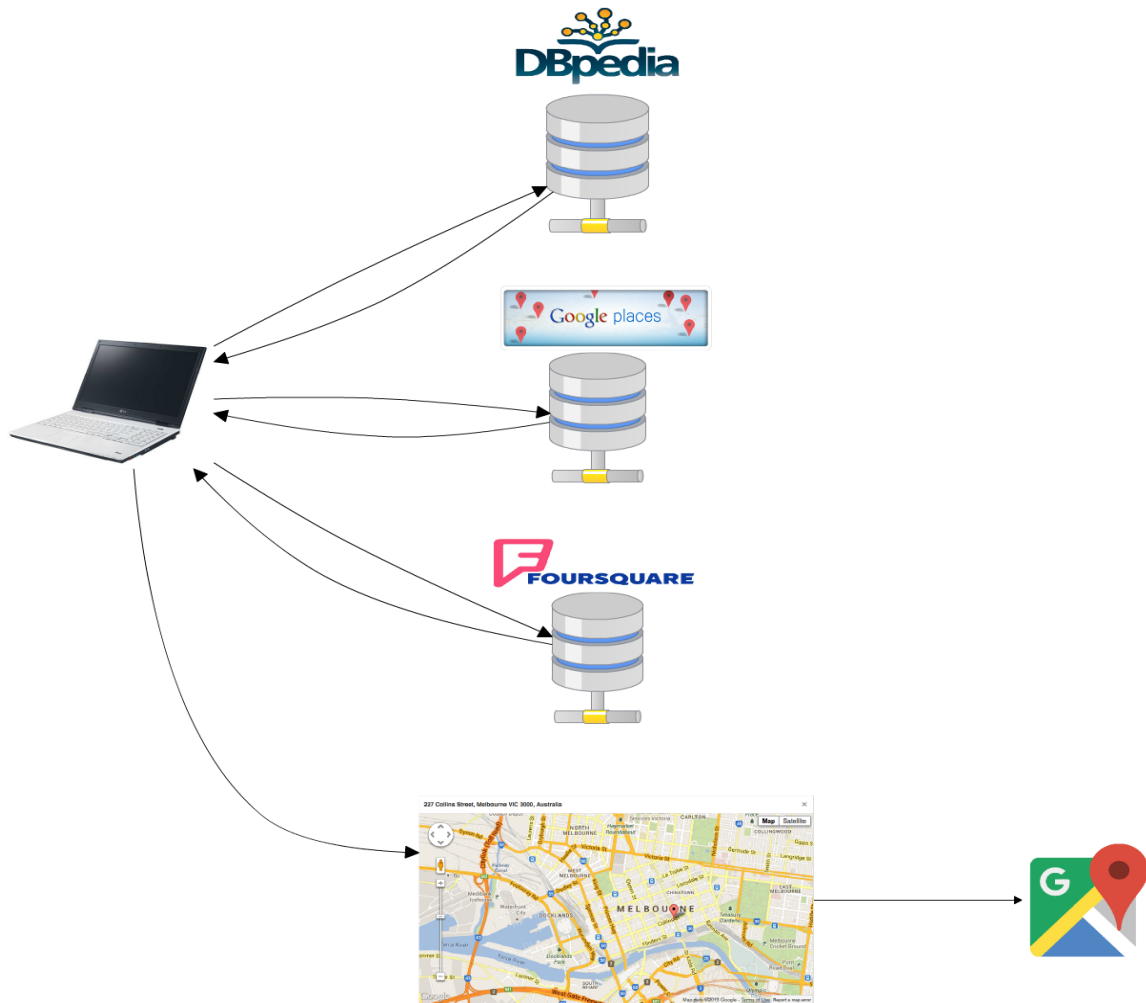


Ilustración 14: Diagrama In-Tour Web con otros sistemas

En el presente diagrama es necesario tener en cuenta los siguientes aspectos:

- El usuario cuando realiza una búsqueda, se realizará una consulta a tres bases de datos:
 - DBpedia.
 - Google Places.
 - Foursquare.
- Tras la búsqueda, el usuario seleccionará las características deseadas, las cuales han sido obtenidas como resultado de una consulta a cada base de datos.
- Tras la selección de características, el usuario podrá generar una ruta, por tanto, los puntos que forman la ruta serán marcados en el mapa que nos proporciona el servicio de Google Maps.

3.3.4.3 Características de los usuarios.

En el presente apartado se describirán las características generales de los usuarios del producto, ya que esto será una buena técnica para adquirir los requisitos de software según las necesidades de los mismos.

Para ello, se plantearán diferentes escenarios, los cuales nos permitirán describir todos los perfiles posibles de los usuarios que hagan uso de la aplicación web. De este modo, se empleará la siguiente plantilla para dicho fin:



IDENTIFICACIÓN DEL ESCENARIO	Código de escenario: Nombre de personaje: Descripción del escenario:
CARACTERÍSTICAS	Edad: Profesión/ tareas: Frecuencia de uso: Conocimiento de la tarea: Formación tecnológica: Limitaciones: Razón de uso:
GRUPOS DE USUARIOS POTENCIALES Y NO POTENCIALES CUBIERTOS EN EL ESCENARIO	
PROBLEMAS	
APORTACIONES	

Ilustración 15: Plantilla Escenarios de Perfiles de Usuario

Donde:

- **IDENTIFICACIÓN DEL ESCENARIO:**
 - Código de escenario: Se trata del número identificador del escenario a describir.
 - Nombre de personaje.
 - Descripción del escenario: Descripción de la situación en la que se encuentra el usuario.
- **CARACTERÍSTICAS:**
 - Edad: En este caso tendremos diversas categorías:
 - Infantil: De 0 a 12 años.
 - Adolescente: De 12 a 18 años.
 - Joven: De 18 a 35 años.
 - Adulto: De 36 a 65 años.
 - Anciano: A partir de los 65 años.
 - Profesión/tareas: Estudiante, Empleado, Empresario, Ninguna, Otras...etc.

- Frecuencia de uso: Habitual, Circunstancial, Ocasional, Ninguna, etc.
 - Conocimiento de la tarea: Alto, Medio, Bajo.
 - Formación tecnológica: Alta, Media, Baja.
 - Razón de uso: Educativo, Recreativa, Curiosidad, Necesidad, Otras...etc.
- **GRUPOS DE USUARIOS POTENCIALES Y NO POTENCIALES CUBIERTOS EN EL ESCENARIO:** Se mencionarán los grupos y los tipos de personas a los que les puede afectar el escenario.
 - **PROBLEMAS:** Se refiere a aquellos problemas o dificultades que pueden encontrar los usuarios al hacer uso de la aplicación web.
 - **APORTACIONES:** Indica la manera en la que la aplicación web cubrirá una necesidad dada en ese escenario.

En base a lo comentado anteriormente, se procede a mostrar los diferentes escenarios analizados:



IDENTIFICACIÓN DEL ESCENARIO	Código de escenario: 01 Nombre de personaje: Raquel Descripción del escenario: Raquel es una adolescente de 14 años que desea realizar un viaje con sus padres por vacaciones. Quiere darles una sorpresa y planificar el viaje ella sola a pesar de tener un desconocimiento completo sobre como planificar un viaje que cubra los gustos e intereses de la familia.
CARACTERÍSTICAS	Edad: Adolescente, 14 años. Profesión/ tareas: Estudiante de Secundaria. Frecuencia de uso: Ninguna. Conocimiento de la tarea: Bajo. Formación tecnológica: Alta. Limitaciones: Desconocimiento de la planificación de viajes. Razón de uso: Necesidad por aprender a planificar viajes.
GRUPOS DE USUARIOS POTENCIALES Y NO POTENCIALES CUBIERTOS EN EL ESCENARIO	Usuarios con desconocimiento del sector turístico y que desean planificar viajes de una manera sencilla.
PROBLEMAS	El usuario puede ser demasiado joven para realizar una buena planificación de su viaje debido a la falta de conocimiento en el sector turístico.
APORTACIONES	A través de la aplicación web, el buscador le permitirá buscar por cualquier tipo de entidad lo cual ampliará el área de búsqueda y le permitirá elegir de una manera sencilla aquellas características que desea que se relacionen con los lugares a visitar.

Tabla 24: Escenario Perfiles Usuario 01



IDENTIFICACIÓN DEL ESCENARIO	Código de escenario: 02 Nombre de personaje: David Descripción del escenario: David es un joven de 25 años que desea vivir nuevas experiencias viajando por todo el mundo al mismo tiempo que adquirir conocimientos sobre las culturas de otros países, su historia y visitar los monumentos más destacados.
CARACTERÍSTICAS	Edad: Joven, 25 años. Profesión/ tareas: Profesor. Frecuencia de uso: Habitual. Conocimiento de la tarea: Alto. Formación tecnológica: Alta. Limitaciones: Ninguna. Razón de uso: Educativa y Recreativa.
GRUPOS DE USUARIOS POTENCIALES Y NO POTENCIALES CUBIERTOS EN EL ESCENARIO	Usuarios que necesitan salir de la rutina diaria y tienen interés por conocer otras costumbres y culturas que hasta el momento han tenido desconocimiento de las mismas.
PROBLEMAS	Aparentemente ninguno.
APORTACIONES	A través de la aplicación web, el usuario podrá generar rutas a partir de la combinación de diferentes características culturales e históricas, permitiéndole planificar un viaje en el cual visitar lugares en relación a las mismas.

Tabla 25: Escenario Perfiles Usuario 02



IDENTIFICACIÓN DEL ESCENARIO	Código de escenario: 03 Nombre de personaje: Eduardo Descripción del escenario: Eduardo es un hombre adulto de 44 años el cual tiene como hobby viajar y realizar una valoración sobre cada viaje que emprende con el fin de mostrar su satisfacción/insatisfacción ante el mismo.
CARACTERÍSTICAS	Edad: Adulto, 40 años. Profesión/ tareas: Estadístico. Frecuencia de uso: Ocasional. Conocimiento de la tarea: Alto. Formación tecnológica: Alta. Limitaciones: Ninguna. Razón de uso: Recreativa.
GRUPOS DE USUARIOS POTENCIALES Y NO POTENCIALES CUBIERTOS EN EL ESCENARIO	Usuarios que ven imprescindible aportar su opinión con fines estadísticos en un futuro.
PROBLEMAS	Aparentemente ninguno.
APORTACIONES	A través de la aplicación web, el usuario podrá aportar su opinión mediante la opción de valoración de rutas tras la generación de cada una de ellas. Es necesario que se puedan valorar rutas completas en lugar de valorar uno por uno cada lugar que forma parte de ella.

Tabla 26: Escenario Perfiles Usuario 03



IDENTIFICACIÓN DEL ESCENARIO	Código de escenario: 04 Nombre de personaje: Eulalia Descripción del escenario: Eulalia es una anciana de 73 años que desea realizar habitualmente pequeñas rutas turísticas con sus nietos con el fin de enseñarles y que entiendan mejor aquello que estudian en la asignatura de Historia y Geografía.
CARACTERÍSTICAS	Edad: Anciana, 75 años. Profesión/ tareas: Jubilada. Frecuencia de uso: Habitual. Conocimiento de la tarea: Alto. Formación tecnológica: Baja. Limitaciones: Uso de la tecnología. Razón de uso: Educativa y Curiosidad.
GRUPOS DE USUARIOS POTENCIALES Y NO POTENCIALES CUBIERTOS EN EL ESCENARIO	Usuarios que tienen mucho tiempo libre y que consideran el turismo una manera de entretenimiento. Además, se trata de usuarios con poca formación tecnológica debido a que no han adquirido un conocimiento previo.
PROBLEMAS	Dificultades a la hora de usar cualquier dispositivo y poca experiencia en el uso de tecnología y servicios ofrecidos.
APORTACIONES	La aplicación web deberá mostrarse de una manera sencilla e intuitiva a la hora de realizar las operaciones sobre la misma indicando de manera clara los pasos que es necesario realizar para obtener las rutas deseadas.

Tabla 27: Escenario Perfiles Usuario 04

3.3.4.4 Restricciones

- La aplicación web debe ser utilizada mediante conexión a Internet.
- Lenguaje de programación: Java.
- Bases de datos empleadas: DBpedia, Google Places y Foursquare.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema deberá tener un diseño intuitivo y una implementación sencilla.

3.3.5 Especificación de Requisitos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir diseñar un sistema que satisfaga estos requisitos, y que permita planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas.

Con la finalidad de especificar la mayor cantidad de información importante sobre los requisitos, se hará uso de la siguiente plantilla para todos los requisitos:

Identificador: TIPO-XX	
Título	Título del requisito
Proviene de	Caso/s de uso de los que proviene el requisito.
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Descripción del requisito.

Tabla 28: Plantilla de requisito de software

Los campos recogidos en la anterior tabla tiene el siguiente significado:

- **Identificador:** Permite identificar unívocamente cada requisito en función de la codificación TIPO-XX:
 - **TIPO:** Se refiere al tipo de requisito. En este caso, como se especificarán los requisitos funcionales y no funcionales, tomará los valores RSF o RSNF, respectivamente.
 - **XX:** Hace referencia al número identificador del requisito.
- **Título:** Título asignado al requisito.
- **Proviene de:** Especifica los casos de uso de los que proviene.
- **Prioridad:** Especifica en qué fase del diseño técnico de la aplicación se debe incluir la funcionalidad de dicho requisito.
- **Estabilidad:** Indica la posibilidad de que un requisito pueda ser modificado de acuerdo con las impresiones del cliente sobre dicho requisito o con el diseño técnico.
- **Descripción:** Se realiza una explicación acerca de la funcionalidad o restricción que el requisito va a implicar en el proyecto.

3.3.5.1 Requisitos Funcionales

Identificador: RSF-01	
Título	Búsqueda por entidad
Proviene de	CU-01
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Cada búsqueda se realizará de acuerdo a la siguiente clasificación: -PERSONA. -LUGAR. -ORGANIZACIÓN. -EVENTO. -OTROS.

Tabla 29: Requisito RSF-01

Identificador: RSF-02	
Título	Características de entidad DBpedia
Proviene de	CU-01
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Cada búsqueda deberá obtener un mínimo de 0 y un máximo de 7 características diferentes si la búsqueda se realizó a partir de la base de datos DBpedia.

Tabla 30: Requisito RSF-02

Identificador: RSF-03	
Título	Características de entidad Google Places
Proviene de	CU-01
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Cada búsqueda deberá obtener un mínimo de 0 y un máximo de 5 características diferentes si la búsqueda se realizó a partir de la base de datos Google Places.

Tabla 31: Requisito RSF-03

Identificador: RSF-04	
Título	Características obtenidas tras búsqueda
Proviene de	CU-01, CU-02
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema mostrará las características obtenidas por cada búsqueda en la sección “Resultados Obtenidos”.

Tabla 32: Requisito RSF-04

Identificador: RSF-05	
Título	Características seleccionadas
Proviene de	CU-01, CU-02
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema mostrará en la sección “Aspectos Seleccionados” las características que fueron seleccionadas en la sección “Resultados Obtenidos”.

Tabla 33: Requisito RSF-05

Identificador: RSF-06	
Título	Características seleccionadas 2
Proviene de	CU-01, CU-02
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema no mostrará las características seleccionadas en la sección "Resultados Obtenidos".

Tabla 34: Requisito RSF-06

Identificador: RSF-07	
Título	Características descartadas
Proviene de	CU-01, CU-02, CU-03
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema mostrará las características descartadas en la sección "Resultados Obtenidos".

Tabla 35: Requisito RSF-07

Identificador: RSF-08	
Título	Características descartadas 2
Proviene de	CU-01, CU-02, CU-03
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema no mostrará las características descartadas en la sección "Aspectos Seleccionados".

Tabla 36: Requisito RSF-08

Identificador: RSF-09	
Título	Características seleccionadas para generación de ruta
Proviene de	CU-01, CU-02, CU-04
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El usuario deberá seleccionar 2 o más características para poder generar una ruta.

Tabla 37: Requisito RSF-09

Identificador: RSF-10	
Título	Combinación de características entre entidades
Proviene de	CU-01, CU-02
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El usuario podrá seleccionar características con relación a diferentes entidades.

Tabla 38: Requisito RSF-10

Identificador: RSF-11	
Título	POIs de las características
Proviene de	CU-04
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema obtendrá una localización o POI asociada a cada característica durante el proceso de generación de ruta.

Tabla 39: Requisito RSF-11

Identificador: RSF-12	
Título	Visualización de POIs en el mapa
Proviene de	CU-04
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema mostrará y localizará en el mapa los POIs pertenecientes a la ruta generada.

Tabla 40: Requisito RSF-12

Identificador: RSF-13	
Título	Información de cada POI
Proviene de	CU-04
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	<p>El sistema mostrará la siguiente información en relación a cada POI cuando el usuario pulse sobre el mismo:</p> <ul style="list-style-type: none"> • Nombre de la característica. • Latitud. • Longitud. • Dirección. • Fuente. • Imagen.

Tabla 41: Requisito RSF-13

Identificador: RSF-14	
Título	Valoración de cada ruta
Proviene de	CU-05
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema almacenará la valoración de cada ruta realizada por el usuario.

Tabla 42: Requisito RSF-14

Identificador: RSF-15	
Título	POIs alrededor
Proviene de	CU-04
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	A través de localizaciones obtenidas mediante consultas DBpedia y Google Places, el sistema obtendrá puntos de interés alrededor de cada localización en un radio máximo de 10000 metros realizando consultas a las bases de datos Google Places y Foursquare.

Tabla 43: Requisito RSF-15

Identificador: RSF-16	
Título	Número de POIs alrededor
Proviene de	CU-04
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Cada búsqueda realizada a Google Places y Foursquare deberá obtener un máximo de 5 puntos de interés (POIs) alrededor de una localización dada.

Tabla 44: Requisito RSF-16

3.3.5.2 Requisitos No Funcionales

Identificador: RSNF-01	
Título	Interfaz adaptativa
Proviene de	RU-13
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadoras personales, dispositivos tableta y teléfonos inteligentes.

Tabla 45: Requisito RSNF-01

Identificador: RSNF-02	
Título	Usuarios expertos y novatos
Proviene de	Ninguno
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El tiempo de aprendizaje del sistema por un usuario será menor a 1 hora.

Tabla 46: Requisito RSNF-02

Identificador: RSNF-03	
Título	Enlaces identificativos
Proviene de	Ninguno
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Los enlaces deben identificar claramente su destino incluso cuando se leen fuera de su contexto.

Tabla 47: Requisito RSNF-03

Identificador: RSNF-04	
Título	Idioma
Proviene de	RU-12
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El idioma de la interfaz será en Castellano.

Tabla 48: Requisito RSNF-04

Identificador: RSNF-05	
Título	Tasa de errores por parte del usuario
Proviene de	Ninguno
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.

Tabla 49: Requisito RSNF-05

Identificador: RSNF-06	
Título	Tasa de errores por parte del usuario
Proviene de	Ninguno
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema debe proporcionar mensajes de error o aviso que sean informativos y orientados al usuario final.

Tabla 50: Requisito RSNF-06

3.4 MATRIZ DE TRAZABILIDAD.

Durante el ciclo de vida del desarrollo software de un proyecto, una de las fases más importantes es la especificación de requisitos de software. Por tanto, para la obtención de los mismos como bien comentamos anteriormente se realiza a partir de los casos de uso y los requisitos de usuario.

De este modo, para comprobar la procedencia de cada requisito funcional con los casos de uso definidos, a continuación se mostrará la matriz de trazabilidad que recoge dicha correspondencia mencionada:

	CU-01	CU-02	CU-03	CU-04	CU-05
RSF-01	X				
RSF-02	X				
RSF-03	X				
RSF-04	X	X			
RSF-05	X	X			
RSF-06	X	X			
RSF-07	X	X	X		
RSF-08	X	X	X		
RSF-09	X	X		X	
RSF-10	X	X			
RSF-11				X	
RSF-12				X	
RSF-13				X	
RSF-14					X
RSF-15				X	
RSF-16				x	

Tabla 51: Matriz de Trazabilidad Requisitos Funcionales vs Casos de Uso

4 DISEÑO

En el presente apartado se determina de manera general como funcionará el sistema. Según Pressman: “El diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo”. De este modo, para llevar a cabo la definición de la fase de diseño del proyecto se empleará el modelo 4+1 [8].

El modelo 4+1 fue diseñado por Philippe Kruchten, empleado para describir la arquitectura de sistemas software, basados en el uso de múltiples puntos de vista.

Una vista no es más que una representación de todo el sistema software desde una determinada perspectiva, y un punto de vista se define como un conjunto de reglas (o normas) para realizar y entender las vistas.

Por consiguiente, lo que propone Kruchten es que un sistema software se ha de documentar y mostrar (tal y como se propone en el estándar IEEE 1471-2000) con 4 vistas bien diferenciadas y estas 4 vistas se han de relacionar entre sí con una vista más, que es la denominada vista “+1”. Estas 4 vistas las denominó Kruchten como:

- Vista Lógica.
- Vista de Desarrollo.
- Vista de Procesos.
- Vista Física

Y vista “+1”, que tiene función de relacionar las 4 vistas citadas, a la denominó vista de escenario.

Cada una de estas vistas ha de mostrar toda la arquitectura del sistema software que se esté documentando, pero cada una de ellas ha de documentarse de forma diferente y ha de mostrar distintos aspectos del sistema software.

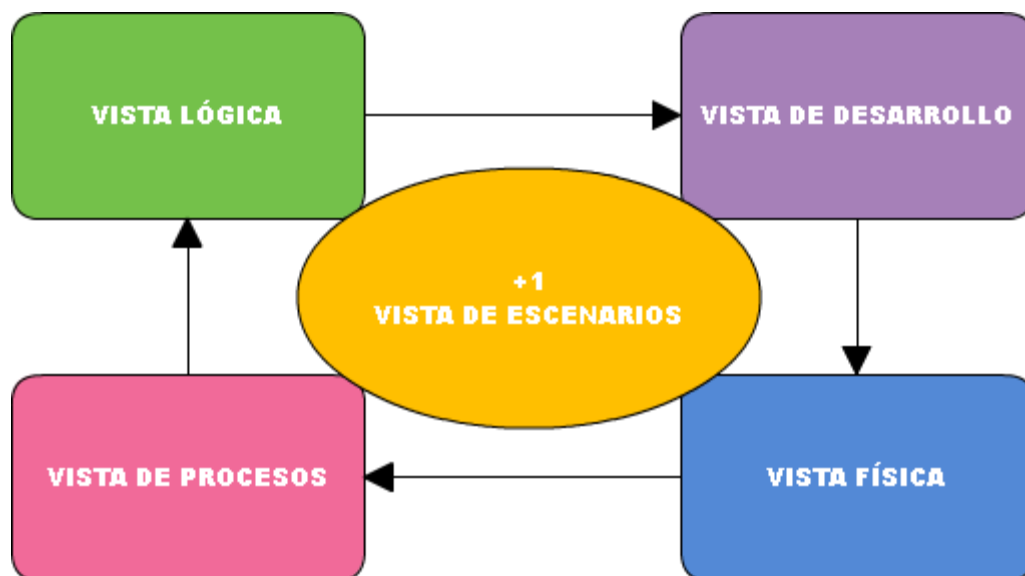


Ilustración 16: Modelo 4+1

4.1 VISTA LÓGICA

La vista lógica está enfocada en describir la estructura y funcionalidad del sistema. En la misma se representa la funcionalidad que el sistema proporcionará a los usuarios finales, es decir, se realiza una representación sobre lo que el sistema debe hacer, y las funciones y servicios que ofrece. Para ello, se hará uso de un diagrama de clases.

El diagrama de clases es un tipo de diagrama de estructura estática la cual es definida mostrando sus clases, atributos, operaciones (o métodos), y las relaciones entre los objetos. En este caso, al prever que se emplearán un gran número de clases durante la fase de implementación, el diagrama realizado consta de las principales clases que va a constituir al mismo con el fin de simplificar su visualización.

De este modo, el diagrama resultante de la fase de análisis y a partir del cual se toman las decisiones de diseño de las clases software es el que se muestra a continuación:

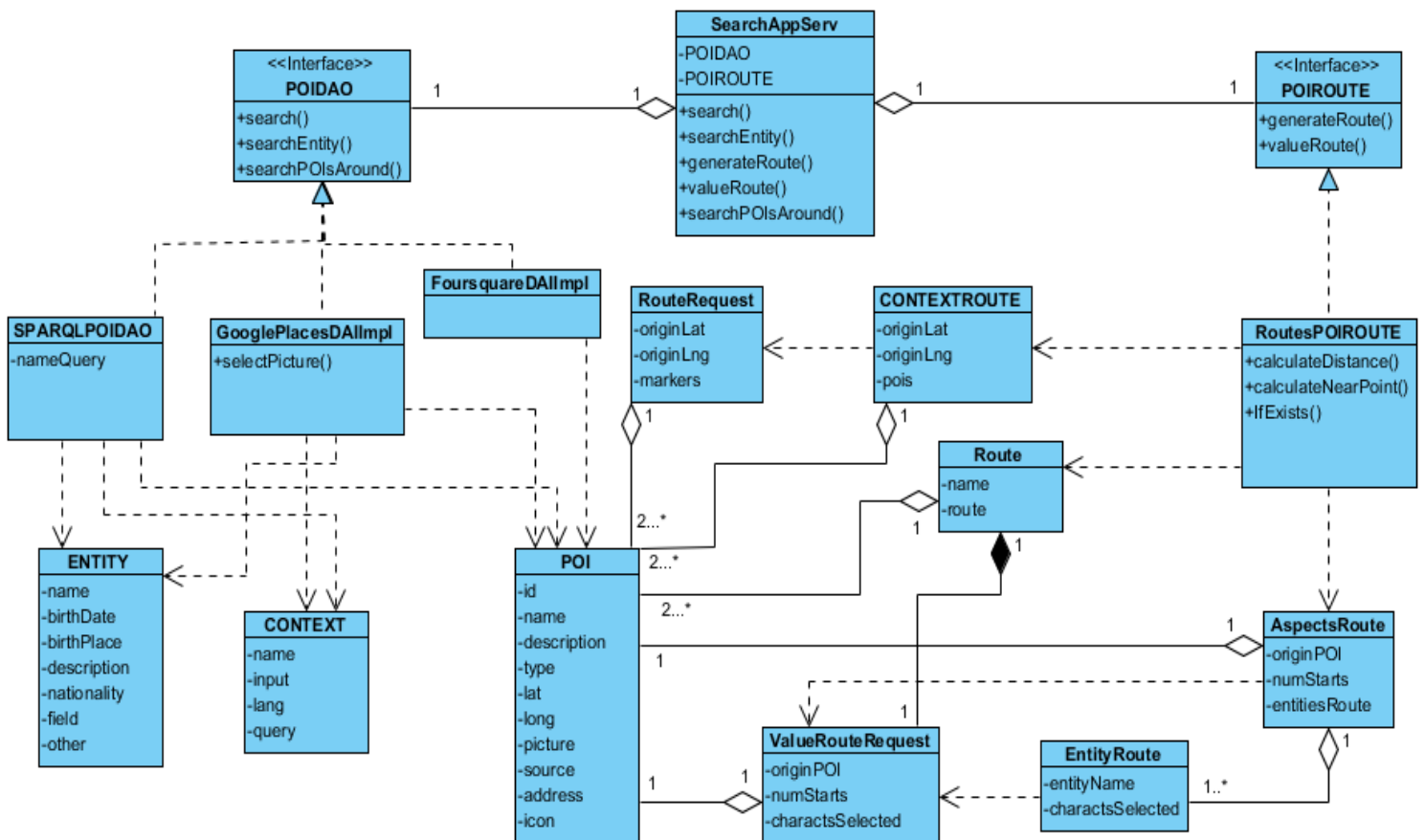


Ilustración 17: Diagrama de clases

4.1.1 Descripción General del Diagrama de Clases

Como podemos observar en el anterior diagrama, la aplicación web estará implementada en varias capas. De este modo, cada vez que el usuario realice cualquier acción, las órdenes se enviarán en primer lugar a la clase llamada *SearchAppServ*.

La clase *SearchAppServ* consta de las principales funcionalidades de la aplicación:

- *searchEntity()*: Función que permitirá buscar por la entidad indicada por parte del usuario y devolverá como resultado un conjunto de características en relación a dicha entidad.
- *search()*: Función que permitirá obtener una localización (latitud, longitud) a través de cada característica seleccionada por el usuario, con el fin de obtener puntos de interés que permitan generar la ruta posteriormente.
- *generateRoute()*: Función que generará una ruta en base a las localizaciones obtenidas a partir de las características seleccionadas por el usuario.
- *valueRoute()* : Función que permitirá que el usuario valore cada ruta generada.
- *searchPOIsAround()*: Función que permitirá obtener otros puntos de interés dada una localización en un determinado radio de distancia.

Por consiguiente, la clase *SearchAppServ* tendrá una relación de agregación con las interfaces *POIDAO* Y *POIROUTE*, que dependiendo de qué orden haya sido indicada por parte del usuario entonces se encargará de su procesamiento una de ellas. Por tanto, dependiendo de la función a realizar se dispondrá de dos cauces distintos a tomar:

- *POIDAO*: Interfaz que especifica los métodos *searchEntity()*, *search()* y *searchPOIsAround()*.
- *POIROUTE*: Interfaz que especifica los métodos *generateRoute()* y *valueRoute()*.

De este modo, la implementación de dichas interfaces será llevado a cabo por distintas clases. Para el caso de la interfaz *POIDAO*, la misma será implementada mediante las clases *SPARQLPOIDAO*, *GooglePlacesDAImpl* y *FoursquareDAImpl* que se corresponden con el acceso a la base de datos DBpedia, Google Places y Foursquare respectivamente.

Tanto *SPARQLPOIDAO* como *GooglePlacesDAImpl* tendrán una relación de dependencia con las siguientes clases:

- *ENTITY*: La presente clase representará la entidad buscada por el usuario. La misma estará constituida por el conjunto de características que describen a aquello buscado como nombre, fecha de nacimiento, descripción, área profesional y otros.
- *CONTEXT*: Dicha clase representa del contexto al que se accederá para llevar a cabo el proceso de las consultas, como por ejemplo en qué idioma se buscarán las características y el nombre de la entidad buscada, entre otros.
- *POI*: Se trata de la clase que constituye a cada punto de interés que será representado en el mapa y a partir de los cuales se crearán las rutas. La información que contiene cada punto de interés será de gran importancia ya que gran parte de la misma será mostrada al usuario cuando el mismo pulse sobre cada POI que aparezca en el mapa. Dicha información será la siguiente:
 - *id*: Se trata del identificador del punto de interés.

- *name*: Nombre de la característica a partir de la cual se obtuvo el POI.
- *description*: Descripción del lugar.
- *type*: Tipo de POI, ya sea una persona, organización, lugar, evento...etc.
- *lat*: Latitud del punto de interés.
- *long*: Longitud del punto de interés.
- *picture*: Enlace a la imagen asociada al POI.
- *source*: Fuente de la que proviene, es decir, si el punto de interés ha sido encontrado a través de la base de datos DBpedia, Google Places o Foursquare.
- *address*: Dirección del POI.
- *icon*: Nombre de la imagen que servirá como marker en el mapa, ya que cada punto de interés tendrá un marker diferente según el tipo de característica del que proviene.

En el caso de *FoursquareDAImpl*, solamente tiene una relación de dependencia con la clase POI, ya que hará uso de este tipo de objeto en el proceso de obtener puntos de interés alrededor de cada localización mediante un determinado radio de distancia. Dicha función se corresponde con el método *searchPOIsAround()*. Sin embargo, los métodos *search()* y *searchEntity()* quedan como “no soportados” en esta clase ya que a partir de esta base de datos no se obtienen características dada una entidad o una localización a partir de una característica.

Por otro lado, la interfaz *POIROUTE* será implementada mediante la clase *RoutesPOIROUTE*, la cual se encargará tanto de generar las rutas como de almacenar la valoración de las mismas por parte del usuario.

En el caso de llevar a cabo la generación de rutas, la clase *RoutesPOIROUTE* tendrá una relación de dependencia con las siguientes clases:

- *CONTEXTROUTE*: La presente clase representará el contexto al que accederá la clase *RoutesPOIROUTE* para realizar la generación de cada ruta. Representa los datos necesarios para la construcción de la misma, los cuales son los siguientes:
 - *originLat*: Representa la latitud del punto de origen en la ruta.
 - *originLng*: Representa la longitud del punto de origen en la ruta.
 - *pois*: Son los puntos de interés generados a partir de las localizaciones (latitud, longitud) obtenidas mediante las características seleccionadas por parte del usuario.

Estos datos son obtenidos a partir de la clase *RouteRequest*, con la cual tiene dependencia *CONTEXTROUTE*. Aparentemente, parecen contener ambas clases los mismos datos, sin embargo, es necesario procesar los datos provenientes de un objeto de tipo *RouteRequest* mapeándolos a un tipo adecuado, puesto que al emplear los servicios REST es más difícil manipular los datos recibidos a partir del servicio.

Además, es necesario mencionar que *RouteRequest* y *CONTEXTROUTE* tienen una asociación de agregación con la clase POI debido a que los datos descritos y por las que están constituidos ambas son objetos de dicha clase.

- *Route*: Se trata de la clase que representará la ruta obtenida y que estará constituida por los siguientes atributos:

- *name*: Se trata del nombre asignado a la ruta.
- *route*: Es el conjunto de puntos de interés que constituyen la ruta generada, en el orden que deben ser visitados mediante el criterio de la mínima distancia entre puntos. También, es preciso mencionar que este conjunto de puntos de interés son objetos de tipo *POI*, por tanto, la clase *Route* tendrá una relación de dependencia con la clase *POI*.

Respecto a la funcionalidad de almacenar las valoraciones llevadas a cabo mediante los usuarios por cada ruta generada, es necesario mencionar que la clase *RoutesPOIRoute* tiene una dependencia con la clase *AspectsRoute*, la cual llevará a cabo este procedimiento.

Para llevar a cabo el almacenamiento de las rutas, primero es preciso indicar de qué manera se llevará a cabo. De este modo, se almacenará las estrellas marcadas por medio del usuario (De 0 a 5 estrellas) junto con las características que fueron seleccionados por el usuario con el fin de regenerar la ruta en un futuro. Además, dichas características irán acompañadas de la entidad a partir de la cual se obtuvieron.

Por consiguiente, un objeto de tipo *AspectsRoute* estará constituido por los siguientes atributos:

- *originPOI*: Se trata del punto de origen de la ruta generada. Como se trata de un objeto de tipo *POI*, la clase *AspectsRoute* tendrá una asociación de agregación con la clase *POI*.
- *numStarts*: Número de estrellas seleccionadas por el usuario para realizar la valoración de la ruta.
- *entitiesRoute*: Se trata de un conjunto de objetos de tipo *EntityRoute*, por tanto, *AspectsRoute* tiene una asociación de agregación con dicha clase. La clase *EntityRoute*, se define mediante los siguientes atributos:
 - *entityName*: Nombre de la entidad a partir de la cual se obtuvieron algunas de las características seleccionadas por el usuario y que permitieron generar la ruta.
 - *charactsSelected*: Es el conjunto de características seleccionadas y que proceden de la búsqueda de *entityName*.

Además, es necesario mencionar que los valores de los atributos pertenecientes a la clase *AspectsRoute* y *EntityRoute* son proporcionados mediante un objeto de tipo *ValueRouteRequest*, siendo los datos de este tipo de objeto aquellos que han sido enviados mediante el servicio REST y que, por tanto, es necesario que sean procesados antes de su tratamiento para el almacenamiento de la valoración de rutas. Por este motivo, las clases *AspectsRoute* y *EntityRoute* tienen una relación de dependencia con *ValueRouteRequest*. Por otro lado, *ValueRouteRequest* tiene una asociación de composición con *Route* debido a que para que existan los datos empleados para la valoración debe existir una ruta.

4.1.2 Tarjetas CRC

Las tarjetas CRC son una herramienta de brainstorming, usado como metodología para el diseño de software orientado a objetos creado por Kent Beck y Ward Cunningham las cuales tienen las siguientes características:

- Es una técnica para la representación de sistemas OO.
- Son un medio de comunicación entre diferentes participantes.
- La lluvia de ideas es una buena práctica para sugerir cómo rellenar las tarjetas.
- Permite ver las clases como algo más que depósito de datos, sino conocer el comportamiento de cada una en un alto nivel.

De este modo, tras una explicación general del diagrama de clases se ha considerado una herramienta útil emplear tarjetas CRC con el fin de realizar una mejor especificación de cada una de las clases, las cuales serán mostradas a continuación.

Para la especificación de cada tarjeta CRC, se empleará la siguiente plantilla:

NOMBRE DE LA CLASE/INTERFAZ			
SUPERCLASE			
SUBCLASE			
PROPIEDADES			
RESPONSABILIDADES	Nombre método	Precondiciones	Postcondiciones
COLABORACIONES			

Tabla 52: Plantilla Tarjetas CRC

Donde:

- **Nombre de la clase/ interfaz:** Se trata del nombre de la clase o interfaz asignado en el diagrama de clases.
- **Superclase:** Nombre de la clase de la que extiende.
- **Subclase:** Nombre de las clases que heredan de la misma.
- **Propiedades:** Se seguirá la codificación **Variable: Tipo**
 - **Variable:** Se tratará de uno de los atributos que contiene la clase.
 - **Tipo:** Se trata del tipo de datos del atributo indicado.
- **Responsabilidades:** Se indicarán las funcionalidades más importantes de cada clase, las cuales serán especificadas mediante:
 - **Nombre del método:** Nombre de la función.
 - **Precondiciones:** Condiciones necesarias antes de realizar la función descrita.
 - **Postcondiciones:** Condiciones que se tienen que cumplir una vez realizada la función indicada.
- **Colaboraciones:** Se tratan de las clases que van a ser empleadas para realizar las responsabilidades indicadas.

NOMBRE DE LA CLASE/INTERFAZ	SearchAppServ		
SUPERCLASE	Ninguna		
SUBCLASE	Ninguna		
PROPIEDADES	<ul style="list-style-type: none"> poiDAOs: List<POIDAO> poiRoutes: List<POIROUTE> 		
RESPONSABILIDADES	Nombre método	Precondiciones	Postcondiciones
	searchEntity(Context context)	El usuario deberá introducir en el buscador la entidad deseada	Se invocará al método <i>searchEntity</i> de la interfaz <i>POIDAO</i> para realizar la búsqueda de entidad en las bases de datos DBpedia y Google Places.
	search(Context context)	-El usuario deberá seleccionar 2 o más características. -El usuario deberá pulsar el botón "Generar ruta". -El usuario deberá indicar un punto de origen.	Se invocará al método <i>search</i> de la interfaz <i>POIDAO</i> para obtener localizaciones (latitud, longitud) por cada característica seleccionada.
	generateRoute(ContextRoute contextroute)	-El usuario deberá seleccionar 2 o más características. -El usuario deberá pulsar el botón "Generar ruta". -El usuario deberá indicar un punto de origen.	Se invocará al método <i>generateRoute</i> de la interfaz <i>POIROUTE</i> para generar una ruta.
	valueRoute(AspectsRoute aspects, String numStarts)	Se deberá haber generado una ruta	Se invocará al método <i>valueRoute</i> de la interfaz <i>POIROUTE</i> para almacenar la valoración realizada por el usuario.
	searchPOIsAround(POI poi)	Deberá de disponer de un conjunto de localizaciones extraídas a partir de las características seleccionadas.	Se invocará al método <i>searchPOIsAround</i> de la interfaz <i>POIDAO</i> para obtener POIs alrededor de cada localización a un determinado radio de distancia.
COLABORACIONES	POIDAO, POIROUTE, Context, ContextRoute, AspectsRoute		

Tabla 53: Tarjeta CRC SearchAppServ

NOMBRE DE LA CLASE/INTERFAZ	POIDAO		
SUPERCLASE	Ninguna		
SUBCLASE	Ninguna		
PROPIEDADES	Ninguna		
RESPONSABILIDADES	Nombre método	Precondiciones	Postcondiciones
	searchEntity(Context context)	Deberá invocarse a dicho método desde la clase SearchAppServ	Se especificará la función <i>searchEntity</i> pero no su implementación.
	search(Context context)	Deberá invocarse a dicho método desde la clase SearchAppServ	Se especificará la función <i>search</i> pero no su implementación.
COLABORACIONES	Context, Entity, POI		

Tabla 54: Tarjeta CRC POIDAO

NOMBRE DE LA CLASE/INTERFAZ	SPARQLPOIDAO		
SUPERCLASE	Ninguna		
SUBCLASE	Ninguna		
PROPIEDADES	<ul style="list-style-type: none"> nameQuery: String 		
RESPONSABILIDADES	Nombre método	Precondiciones	Postcondiciones
	searchEntity(Context context)	Ninguna	-Se realizará la implementación del método <i>searchEntity</i> especificado en la interfaz <i>POIDAO</i> . -Se obtendrá un conjunto de características de la entidad a buscar mediante una consulta a la base de datos DBpedia.
	search(Context context)	-Se deberá haber buscado una entidad antes. -Se deberá haber seleccionado 2 o más características.	-Se realizará la implementación del método <i>search</i> especificado en la interfaz <i>POIDAO</i> . -Se obtendrá una localización (latitud, longitud) por cada característica seleccionada mediante consultas a la base de datos DBpedia.
	searchPOIsAround(POI poi)	No soportado	No soportado
COLABORACIONES	Context, Entity, POI		

Tabla 55: Tarjeta CRC SPARQLPOIDAO

NOMBRE DE LA CLASE/INTERFAZ	GooglePlacesDAImpl		
SUPERCLASE	Ninguna		
SUBCLASE	Ninguna		
PROPIEDADES	Ninguna		
RESPONSABILIDADES	Nombre método	Precondiciones	Postcondiciones
	searchEntity(Context context)	Ninguna	-Se realizará la implementación del método <i>searchEntity</i> especificado en la interfaz <i>POIDAO</i> . -Se obtendrá un conjunto de características de la entidad a buscar mediante una consulta a la base de datos Google Places.
	search(Context context)	-Se deberá haber buscado una entidad antes. -Se deberá haber seleccionado 2 o más características.	-Se realizará la implementación del método <i>search</i> especificado en la interfaz <i>POIDAO</i> . -Se obtendrá una localización (latitud, longitud) por cada característica seleccionada mediante consultas a la base de datos Google Places.
	searchPOIsAround(POI poi)	Deberá de disponer de un conjunto de localizaciones extraídas a partir de las características seleccionadas.	-Será invocado por medio del método <i>searchPOIsAround</i> especificado en la interfaz <i>POIDAO</i> . - A partir de localizaciones obtendrá POIs que le puedan interesar al usuario en un determinado radio respecto a cada localización.
COLABORACIONES	Context, Entity, POI		

Tabla 56: Tarjeta CRC GooglePlacesDAImpl

NOMBRE DE LA CLASE/INTERFAZ	FoursquareDAImpl		
SUPERCLASE	Ninguna		
SUBCLASE	Ninguna		
PROPIEDADES	Ninguna		
RESPONSABILIDADES	Nombre método	Precondiciones	Postcondiciones
	searchPOIsAround(POI poi)	Deberá de disponer de un conjunto de localizaciones extraídas a partir de las características seleccionadas.	-Será invocado por medio del método <i>searchPOIsAround</i> especificado en la interfaz <i>POIDAO</i> . - A partir de localizaciones obtendrá POIs que le puedan interesar al usuario en un determinado radio respecto a cada localización.
	search(Context context)	No soportado	No soportado
	searchEntity(Context context)	No soportado	No soportado
COLABORACIONES	POI		

Tabla 57: Tarjeta CRC FoursquareDAImpl

NOMBRE DE LA CLASE/INTERFAZ	CONTEXT		
SUPERCLASE	Ninguna		
SUBCLASE	Ninguna		
PROPIEDADES	<ul style="list-style-type: none"> • lang: String • name: String • input: String • query: String 		
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Representa el contexto al que se accederá para la búsqueda de entidad y de localizaciones a través de características seleccionadas. • Dicho contexto engloba el conjunto de información necesaria para realizar consultas tanto en las bases de datos DBpedia y Google Places. 		
COLABORACIONES	Ninguna		

Tabla 58: Tarjeta CRC CONTEXT

NOMBRE DE LA CLASE/INTERFAZ	ENTITY
SUPERCLASE	Ninguna
SUBCLASE	Ninguna
PROPIEDADES	<ul style="list-style-type: none"> • name: String • birthPlace: String • birthDate: String • description: String • nationality: String • field: String • museum: String • other: String
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Representa el conjunto de características de la entidad buscada.
COLABORACIONES	Ninguna

Tabla 59: Tarjeta CRC ENTITY

NOMBRE DE LA CLASE/INTERFAZ	POI
SUPERCLASE	Ninguna
SUBCLASE	Ninguna
PROPIEDADES	<ul style="list-style-type: none"> • id: String • name: String • description: String • type: String • lat: String • long: String • picture: String • source: String • address: String • icon: String
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Representa cada punto de interés que constituyen una ruta.
COLABORACIONES	Ninguna

Tabla 60: Tarjeta CRC POI

NOMBRE DE LA CLASE/INTERFAZ	POIROUTE		
SUPERCLASE	Ninguna		
SUBCLASE	Ninguna		
PROPIEDADES	Ninguna		
RESPONSABILIDADES	Nombre método	Precondiciones	Postcondiciones
	generateRoute(ContextRoute contextroute)	Deberá invocarse a dicho método desde la clase SearchAppServ	Se especificará la función <i>generateRoute</i> pero no su implementación.
	valueRoute(AspectsRoute aspects, String numStarts)	Deberá invocarse a dicho método desde la clase SearchAppServ	Se especificará la función <i>valueRoute</i> pero no su implementación.
COLABORACIONES	ContextRoute, POI, AspectsRoute		

Tabla 61: Tarjeta CRC POIROUTE

NOMBRE DE LA CLASE/INTERFAZ	RoutesPOIRoute		
SUPERCLASE	Ninguna		
SUBCLASE	Ninguna		
PROPIEDADES	Ninguna		
RESPONSABILIDADES	Nombre método	Precondiciones	Postcondiciones
	generateRoute(ContextRoute contextroute)	Disponer de un conjunto de puntos de interés extraídos a partir de las características seleccionadas.	-Se realizará la implementación del método <i>generateRoute</i> especificado en la interfaz <i>POIRoute</i> . -Se ordenarán los puntos de interés extraídos a partir de las características seleccionadas por parte del usuario, mediante el criterio de la distancia más corta.
	calculateDistance(double originLat, double originLng, double destLat, double destLng)	Ninguna	Se obtendrá como resultado el cálculo de la distancia entre dos puntos.
	calculateNearPoint(List<POI> route, int position, ContextRoute contextroute)	Ninguna	Se obtendrá como resultado el punto más cercano a uno dado.
	ifExists(List<POI> route, String lat, String lng)	Ninguna	Se comprobará si existe un determinado punto de interés en la ruta.
COLABORACIONES	valueRoute(AspectsRoute aspects, String numStarts)	Disponer del conjunto de características que seleccionó el usuario para generar la ruta que el mismo desea valorar	-Se realizará la implementación del método <i>valueRoute</i> especificado en la interfaz <i>POIRoute</i> . - Se almacenará la valoración realizada sobre la ruta generada. -Se almacenarán las características que fueron seleccionadas para generar la ruta, con el fin de poder reconstruirla en un futuro.
	ContextRoute, POI, AspectsRoute, Route		

Tabla 62: Tarjeta CRC RoutesPOIRoute

NOMBRE DE LA CLASE/INTERFAZ	CONTEXTROUTE
SUPERCLASE	Ninguna
SUBCLASE	Ninguna
PROPIEDADES	<ul style="list-style-type: none"> • originLat: double • originLng: double • pois: List<POI>
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Representa el contexto al que se accederá para la generación de rutas. • Dicho contexto engloba el punto de origen y el conjunto de puntos de interés extraídos a través de las características que fueron seleccionadas por el usuario.
COLABORACIONES	POI, RouteRequest

Tabla 63: Tarjeta CRC CONTEXTROUTE

NOMBRE DE LA CLASE/INTERFAZ	RouteRequest
SUPERCLASE	Ninguna
SUBCLASE	Ninguna
PROPIEDADES	<ul style="list-style-type: none"> • originLat: String • originLng: String • markers: List<POI>
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Representa el conjunto de datos recibidos mediante el servicio REST para la generación de rutas. • Dichos datos deben mapearse con un objeto de tipo CONTEXTROUTE para que puedan manipularse más fácilmente en el proceso de generación de rutas.
COLABORACIONES	POI

Tabla 64: Tarjeta CRC RouteRequest

NOMBRE DE LA CLASE/INTERFAZ	Route
SUPERCLASE	Ninguna
SUBCLASE	Ninguna
PROPIEDADES	<ul style="list-style-type: none"> • name: String • route: List<POI>
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Representa el conjunto de puntos de interés ordenados según el criterio de la distancia más corta, es decir, se trata de la ruta generada.
COLABORACIONES	POI

Tabla 65: Tarjeta CRC Route

NOMBRE DE LA CLASE/INTERFAZ	AspectsRoute
SUPERCLASE	Ninguna
SUBCLASE	Ninguna
PROPIEDADES	<ul style="list-style-type: none"> • originPOI: String • numStarts: int • entitiesRoute: List<EntityRoute>
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Representa la información de la que se hará uso para almacenar la valoración de rutas realizadas por el usuario. • Los datos almacenados permitirán en un futuro regenerar la ruta.
COLABORACIONES	EntityRoute

Tabla 66: Tarjeta CRC AspectsRoute

NOMBRE DE LA CLASE/INTERFAZ	ValueRouteRequest
SUPERCLASE	Ninguna
SUBCLASE	Ninguna
PROPIEDADES	<ul style="list-style-type: none"> • originPOI: String • numStarts: String • charactsSelected: List<String>
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Representa el conjunto de datos recibidos mediante el servicio REST para realizar la valoración de rutas. • Dichos datos deben mapearse con un objeto de tipo AspectsRoute para que puedan manipularse más fácilmente en el proceso de almacenar la valoración realizada por parte del usuario.
COLABORACIONES	Ninguna

Tabla 67: Tarjeta CRC ValueRouteRequest

NOMBRE DE LA CLASE/INTERFAZ	EntityRoute
SUPERCLASE	Ninguna
SUBCLASE	Ninguna
PROPIEDADES	<ul style="list-style-type: none"> • entityName: String • charactsSelected: List<String>
RESPONSABILIDADES	<ul style="list-style-type: none"> • Se trata de una clase en la que no se implementará ningún método. • Cada objeto de este tipo representa el conjunto de características seleccionadas mediante el usuario y la entidad a partir de la cual se obtuvieron. • Es necesario ya que se almacenará un conjunto de objetos de este tipo con el fin de regenerar la ruta en un futuro.
COLABORACIONES	Ninguna

Tabla 68: Tarjeta CRC EntityRoute

4.2 VISTA DE DESARROLLO

La vista de desarrollo muestra el sistema desde la perspectiva del programador y se ocupa de la gestión del software, es decir, se va a mostrar cómo está dividido el sistema software en componentes y las dependencias que hay entre esos componentes. Para ello, se empleará un diagrama de componentes y un diagrama de paquetes.

4.2.1 Diagrama de Componentes

Un diagrama de componentes tiene como objetivo modelar el sistema o subsistema que será implementado después. De este modo, emplearemos este tipo de diagrama para mostrar los elementos de diseño del sistema de software, permitiendo visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces.

Por consiguiente, los elementos que puede contener serán:

- Componentes.
- Relaciones de dependencia, generalización asociación y realización.
- Paquetes o subsistemas.
- Interfaces.

Por tanto, el diagrama de componentes realizado en función del sistema a desarrollar será el siguiente:

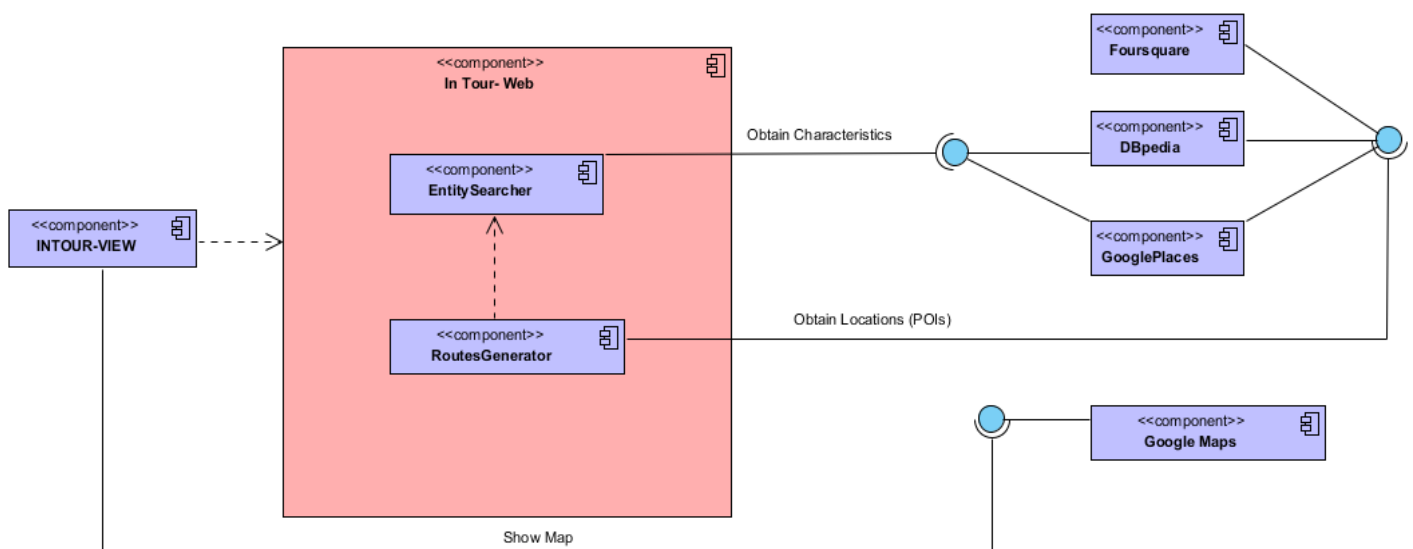


Tabla 69: Diagrama de componentes

En el diagrama anterior podemos observar que los componentes que lo forman están distribuidos siguiendo el modelo de arquitectura Cliente-Servidor en el proceso de comunicación con la aplicación web.

[9] La arquitectura Cliente-Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. De este modo, la red cliente-servidor es una red de comunicaciones en la cual los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son sólo lectura y los que, por el contrario, pueden ser modificados, etc.

Además, los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. A pesar de ello, mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

En este caso, podemos observar que se trata de una arquitectura Cliente-Servidor en tres capas:

- **Cliente:** Se tratan de los usuarios que interactúan con la capa de presentación de la aplicación web. Dicha capa se trata de un software que se encarga de representar los resultados de la aplicación y enviar peticiones al servidor de aplicaciones.
- **Servidor de aplicaciones:** Se trata del software que entrega un resultado útil al usuario. En este caso realizando consultas a diversas bases de datos. Implementa la lógica de negocio estableciendo qué operaciones se pueden realizar sobre el sistema y cómo se llevarán a cabo. De este modo, se encarga de hacer cumplir las reglas de negocio y establecer los procesos de negocio.
- **Servidor de datos:** Organiza y suministra los datos necesarios almacenados en las bases de datos a aquellas aplicaciones con el fin de servir a su lógica de negocio.



Ilustración 18: Arquitectura Cliente-Servidor en 3 capas [9]

Tras la explicación del tipo de arquitectura la cual se va a emplear, se procede a explicar aquellos aspectos en relación al diagrama de componentes realizado. De este modo, se especificarán los distintos componentes que aparecen en el diagrama acorde al tipo de arquitectura comentada:

- **Capa de Presentación (Vista):** Esta capa solamente estará constituida por un único componente llamado *INTOUR VIEW* la cual constituye la unificación de todas las vistas del proyecto. En este caso, solamente se tiene pensado disponer de una vista llamada *Inicio* la cual será el punto de partida al acceder a la aplicación web y en la que se mostrará información sobre la misma. Además, constará de otra vista llamada *Planificador*, a partir de la cual se podrá hacer uso de todas las funcionalidades disponibles y mostrará los resultados obtenidos.
- **Capa del servidor de aplicaciones:** En dicha capa se encontrarán diferentes componentes:
 - *EntitySearcher*: Dicho componente se encargará de capturar el nombre de la entidad introducida por el usuario en el buscador y realizar una consulta a las siguientes base de datos empleadas:

- *DBpedia*.
- *Google Places*.

Las consultas realizadas consistirán en buscar características relacionadas con el nombre buscado. Esto permitirá obtener un mayor número de características de interés para mostrar al usuario tras su búsqueda. Después esto será mostrado al usuario a través del componente *INTOUR VIEW*.

- *RoutesGenerator*: En referencia a dicho componente se puede observar que es aquel encargado de llevar a cabo el proceso de la generación de rutas. Para ello, este llevará a cabo las siguientes funcionalidades:
 - *Obtención de localizaciones*: Tras haber obtenido las características de la entidad buscada, dicho controlador se encarga de realizar una consulta a las bases de datos DBpedia y Google Places con el fin obtener una localización por cada característica y así poder disponer de puntos de interés (POIs) para poder generar la ruta. Además, a partir de dichas localizaciones se obtendrán POIs que le puedan interesar al usuario en un determinado radio respecto a cada localización mediante consultas a las bases de datos Google Places y Foursquare y que también pertenecerán a la ruta que se pretende generar.
 - *Generación de rutas*: Para llevar a cabo dicha funcionalidad será necesario que este disponga de un conjunto de localizaciones y, por tanto, un conjunto de características a partir de las cuales se obtuvieron las mismas. Por dicho motivo, el componente *RoutesGenerator* dependerá de *EntitySearcher* ya que este será el que le suministrará las características de las entidades.
 - *Valoración de rutas*: Dicha funcionalidad se encargará de obtener la valoración asociada a la ruta generada y almacenarla de manera anónima con el fin de tener información en un futuro para un algoritmo de recomendación de tipo filtrado colaborativo ítem-ítem, siendo el uso de este algoritmo opcional en el proceso de implementación.

- **Capa de los servidores de datos:** Como en esta capa se llevará a cabo el proceso de recuperar y almacenar la información, se encontrarán los siguientes componentes, los cuales son servicios externos de los que se hace uso:
 - *DBpedia.*
 - *GooglePlaces.*
 - *Foursquare.*
 - *Google Maps.*

4.2.2 Diagrama de paquetes

Un diagrama de paquetes representa las dependencias entre los paquetes que componen un modelo, es decir, muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones.

Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. Por tanto, estos se encuentran normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los mismos.

A continuación, se mostrará el diagrama de paquetes amoldado a la aplicación *In-Tour Web* permitiendo una mejor organización de las clases durante el proceso de implementación:

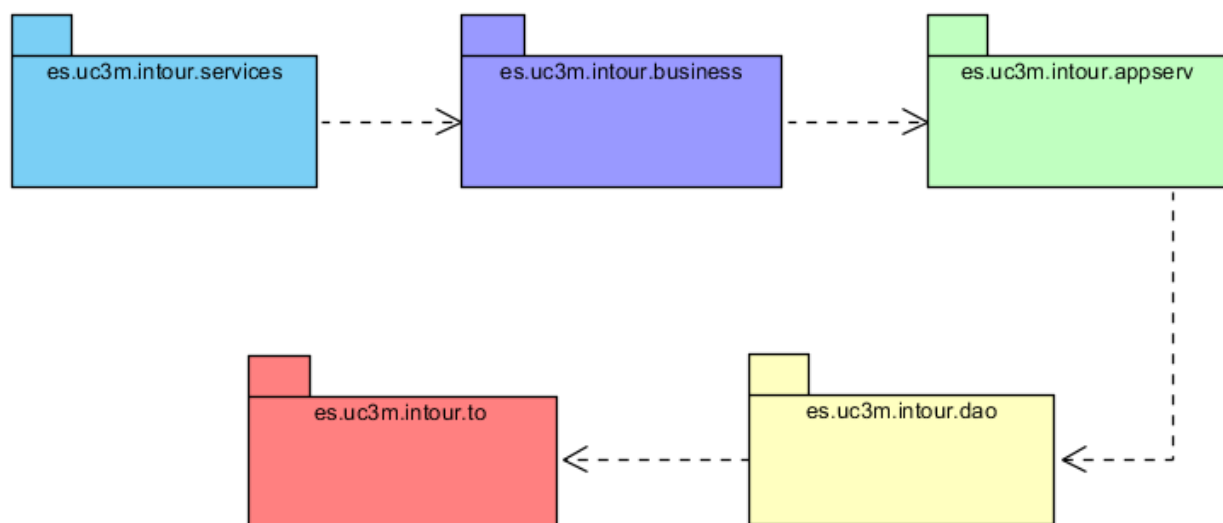


Ilustración 19: Diagrama de componentes

Como podemos observar en el anterior diagrama, se ha decidido agrupar el conjunto de clases en los distintos paquetes que se pueden visualizar. Además, como bien se mencionó anteriormente, la aplicación web estará constituida por varias capas en el proceso de implementación, por tanto, para ejecutar cualquier función es necesario que el ciclo de ejecución vaya pasando por cada una de las capas hasta obtener el resultado.

De este modo, se realizará una pequeña explicación del contenido de cada uno de los paquetes mostrados:

- ***es.uc3m.intour.service***: En dicho paquete se implementarán las clases que representan los servicios que son invocados desde la parte front-end de la aplicación web. Dicho paquete dependerá de *es.uc3m.intour.business*, el cual servirá de pasarela para ejecutar las funciones deseadas.

En este caso, las clases que representan los servicios no han sido mencionadas en el diagrama de clases con el fin de simplificar su construcción, pero el contenido del presente paquete sería el siguiente:

- *SearchEntityService*: Clase que representa el servicio el cual permitirá realizar la búsqueda por entidad.
- *SearchService*: Clase que representa el servicio el cual permitirá realizar la búsqueda de localizaciones (latitud, longitud) por cada característica seleccionada por el usuario.
- *RouteGeneratorService*: Clase que representa el servicio el cual permitirá la generación de rutas.
- *ValueRouteService*: Clase que representa el servicio el cual permitirá la valoración de rutas.

- ***es.uc3m.intour.business***: En el presente paquete se llevará a cabo la especificación de las funciones invocadas desde las clases del paquete *es.uc3m.intour.service* y que, como se ha mencionado antes, servirá de pasarela con el paquete *es.uc3m.intour.appserv* permitiendo invocar a las funciones correspondientes. Por tanto, sirve de enlace entre las funciones invocadas desde el paquete *es.uc3m.intour.service* y las funciones especificadas en el paquete *es.uc3m.intour.appserv*, de modo que el paquete *es.uc3m.intour.business* tendrá una relación de dependencia con el paquete *es.uc3m.intour.appserv*.

En este caso, las clases contenidas en dicho paquete y que actúan como pasarela, no han sido mencionadas en el diagrama de clases al igual que el caso anterior para la simplificación del mismo. Sin embargo, su contenido sería el siguiente:

- *SearchBusiness*: Interfaz que especifica las funciones de la aplicación web.
- *SearchBusinessImpl*: Clase que implementa la interfaz anterior y que invoca a los métodos correspondientes según la función solicitada del paquete *es.uc3m.intour.appserv*.

- ***es.uc3m.intour.appserv***: En dicho paquete se implementará la clase *SearchAppServ* (mencionada en el diagrama de clases), en la cual se invocará a una interfaz u otra dependiendo de la función a realizar, las cuales son *POIDAO* Y *POIROUTE*. Sin embargo, a pesar de poder elegir entre dos cauces distintos en el proceso de ejecución, ambas interfaces van a pertenecer al mismo paquete llamado *es.uc3m.intour.dao*, por tanto, se da una dependencia con el mismo en el diagrama, además de la implementación de otras clases.

- ***es.uc3m.intour.dao***: Se trata del paquete que contendrá aquellas clases que implementarán las funciones invocadas.

De acuerdo a las clases mencionadas en el diagrama de clases, el presente paquete agrupará las siguientes:

- *POIDAO*
- *POIROUTE*
- *SPARQLPOIDAO*
- *GooglePlacesDAImpl*
- *FoursquareDAImpl*
- *RoutesPOIROUTE*

- ***es.uc3m.intour.to***: En este último paquete tiene como contenido aquellas clases que no implementan ningún método sino que son utilizadas durante el proceso de implementación de las funciones en el paquete *es.uc3m.intour.dao*.

De acuerdo a las clases mencionadas en el diagrama de clases, el presente paquete agrupará las siguientes:

- *ENTITY*
- *CONTEXT*
- *POI*
- *CONTEXTROUTE*
- *RouteRequest*
- *Route*
- *AspectsRoute*
- *EntityRoute*
- *ValueRouteRequest*

4.3 VISTA DE PROCESOS

La vista de procesos trata los aspectos dinámicos del sistema, explica los procesos de sistema y cómo se comunican. Se enfoca en el comportamiento del mismo en tiempo de ejecución al igual que este tipo de vista considera aspectos de concurrencia, distribución, rendimiento, escalabilidad,...etc.

Por consiguiente, en esta vista se muestra la manera de comunicación entre los procesos del sistema, es decir, se representa desde la perspectiva de un integrador de sistemas el flujo de trabajo paso a paso de negocio y operaciones de los componentes que conforman el sistema. Para ello, se emplearán diagramas de secuencia por cada operación principal de la aplicación web con el fin de especificar dicha vista del modelo 4+1.

4.3.1 Diagrama de secuencia: Búsqueda por entidad.

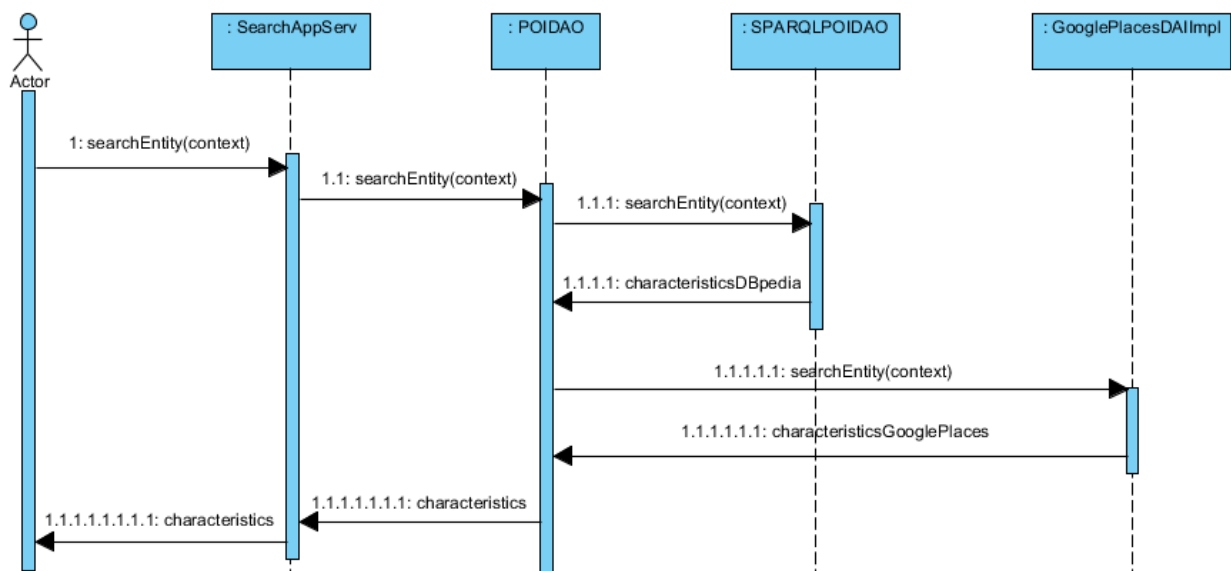


Ilustración 20: Diagrama de secuencia-Búsqueda por entidad

En el anterior diagrama de secuencia se muestran los pasos llevados a cabo durante la ejecución de búsqueda por entidad. En primer lugar, el usuario una vez haya introducido en el buscador el nombre de la entidad, la aplicación web invocará al método *searchEntity* de la clase *SearchAppServ*, luego el método con el mismo nombre de la clase *POIDAO* y a partir de este momento se invocará de nuevo a otro método también con el mismo nombre pero en dos cauces distintos: uno invocando al método de la búsqueda de entidad de *SPARQLPOIDAO* y otro de la clase *GooglePlacesDAImpl*.

Dichos métodos realizarán una consulta a DBpedia y Google Places y se obtendrán un conjunto de características asociadas a la entidad buscada: *characteristicsDBpedia* y *characteristicsGooglePlaces*, respectivamente. Después, en la clase *POIDAO* se obtendrá la unión de ambas características (*characteristics*) y dicho conjunto será enviado a la clase *SearchAppServ* como respuesta a la invocación anteriormente realizada y serán mostradas a través de la interfaz.

4.3.2 Diagrama de secuencia: Generación de rutas.

En el caso de la generación de rutas, será necesario realizar dos diagramas de secuencia para simplificar su representación. En el primer diagrama se mostrarán los pasos llevados a cabo para obtener localizaciones a partir de las características seleccionadas por parte del usuario y en base a dicho resultado se procederá a calcular la ruta mediante dichas localizaciones obtenidas.

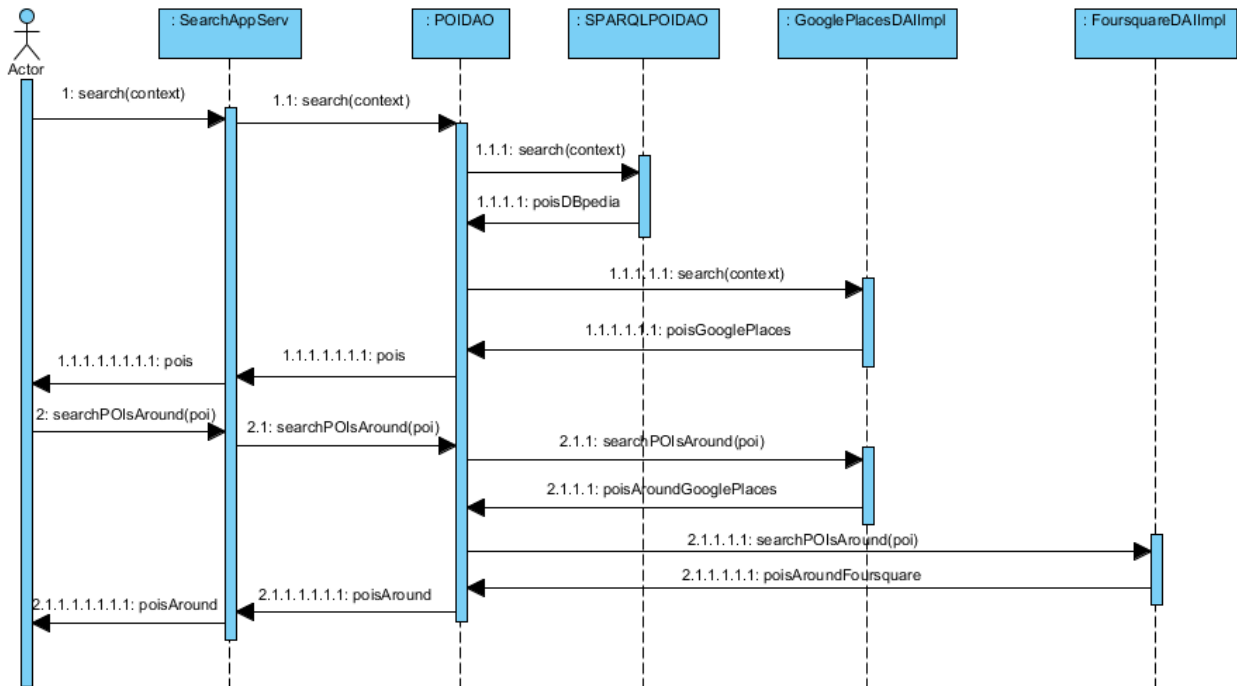


Ilustración 21: Diagrama de secuencia-Obtención de localizaciones

En este primer diagrama, podemos observar que es similar al diagrama de secuencia de búsqueda por entidad mostrado en el apartado anterior. Sin embargo, en este caso se invocará al método *search* el cuál realizará una consulta por cada característica que fue seleccionada por parte del usuario para obtener una localización tanto a las bases de datos DBpedia como Google Places.

El conjunto de localizaciones obtenidas a través de las consultas realizadas (*poiDBpedia* y *poiGooglePlaces*) darán lugar a un conjunto de puntos de interés (*poi*) inicial que se emplearán para extraer nuevas localizaciones.

A partir del conjunto de localizaciones inicial (obtenido mediante las características que fueron seleccionadas), se procede obtener varios puntos de interés por cada localización perteneciente a dicho conjunto a un determinado radio de distancia. Para ello, se invocará al método *searchPOIsAround()* el cual realizará las consultas adecuadas tanto a la base de datos de Google Places como de Foursquare. Esto dará lugar a los puntos obtenidos en los alrededores (*poiAround*).

De este modo, el conjunto total de puntos los cuales serán empleados para generar la ruta será la unión de *poi* y de *poiAround*. Dichas localizaciones serán devueltas a la interfaz pero no serán mostradas puesto que constituirán uno de los parámetros enviados durante la invocación del servicio de generación de rutas.

Una vez se obtuvieron las localizaciones tal y como se especificó en el diagrama de secuencia anterior, se llevará a cabo el proceso de cálculo de la ruta. Como ya disponemos de localizaciones, es necesario ordenar dichos puntos de interés con el fin de obtener una ruta siguiendo el criterio de la distancia más corta entre los puntos. De este modo, a continuación se muestra el diagrama de secuencia del proceso de generación de rutas:

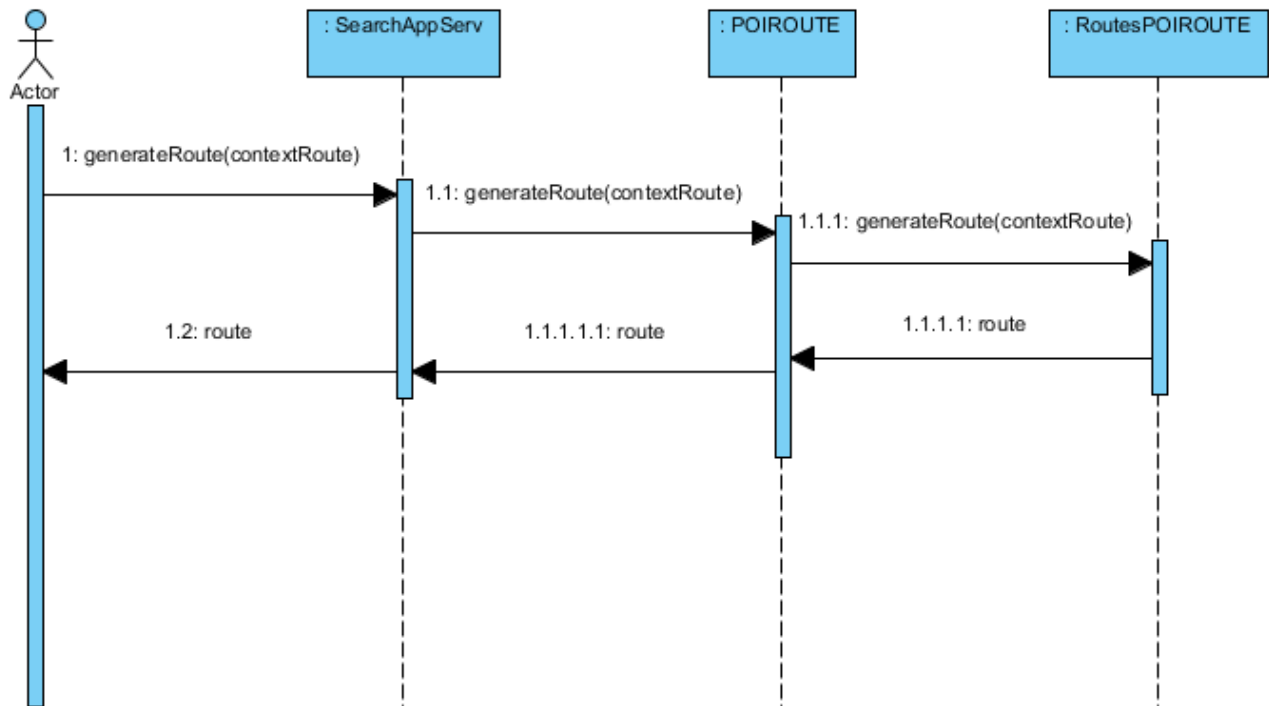


Ilustración 22: Diagrama de secuencia-Generación de rutas

En dicho diagrama, se puede observar que se va invocando al método llamado *generateRoute* a través de las distintas capas en las que estaría estructurada la aplicación web hasta llegar a la clase *RoutesPOIRoute* donde se realizarán los cálculos correspondientes para obtener un conjunto de puntos ordenados, los cuales constituirán la ruta solicitada.

Una vez realizados los cálculos para la obtención de la ruta, la misma es mostrada a través de la interfaz como un listado de puntos ordenados al mismo tiempo que la ruta se deberá mostrar en el mapa.

4.3.3 Diagrama de secuencia: Valoración de rutas

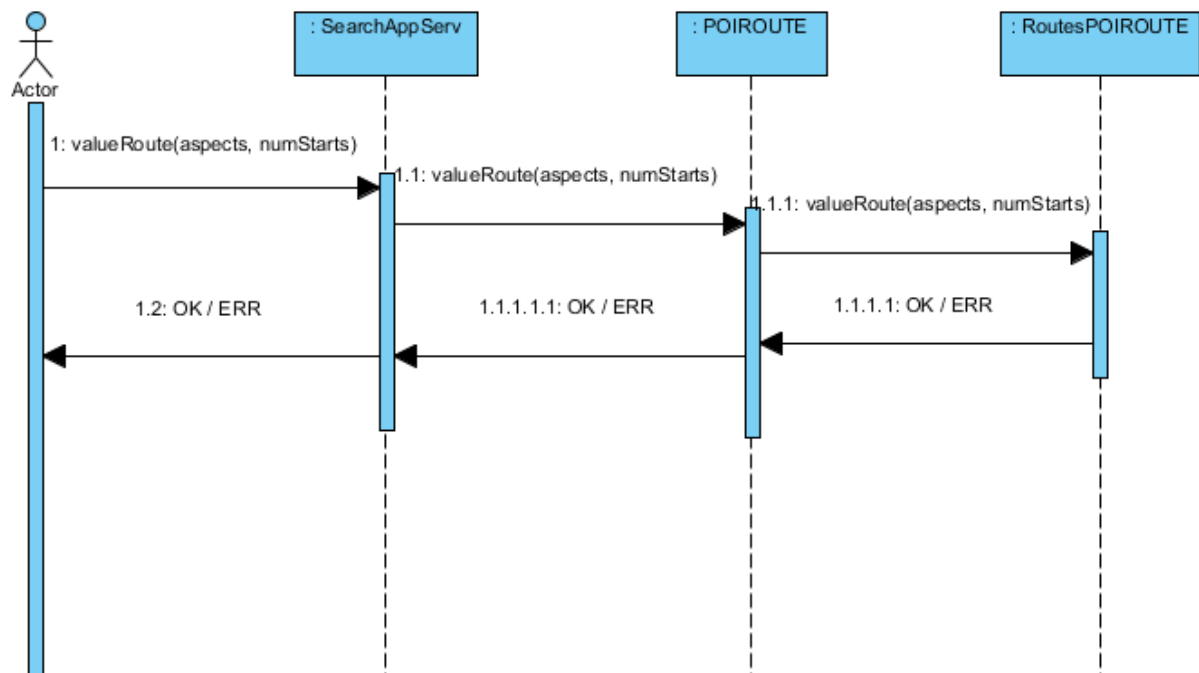


Ilustración 23: Diagrama de secuencia-Valoración de Rutas

En el presente diagrama de secuencia, se muestran los pasos llevados a cabo para realizar la valoración de rutas. De este modo, se invocará al método *valueRoute* a través de las distintas capas en las que estaría estructurada la aplicación web hasta llegar a la clase *RoutesPOIRoute* donde se almacenarán los aspectos de la ruta junto con el número de estrellas que marcó el usuario a través de la interfaz para valorar la ruta generada.

Dichos aspectos de la ruta constituirán el conjunto de características que fueron seleccionadas por el usuario para generar la ruta, donde estará formado por subconjuntos de características asociadas a la entidad que permitió que se obtuvieran.

Una vez almacenada la valoración, se procede a enviar un mensaje de respuesta indicando si el procedimiento se realizó correctamente (*OK*) o por el contrario ocurrió algún tipo de error (*ERR*). Dicho mensaje irá enviándose por todas las capas como resultado de la invocación al método *valueRoute* que se realizó en cada una de ellas, hasta mostrarle al usuario a través de la interfaz un mensaje informativo en consecuencia.

Es preciso volver a mencionar que la ruta será almacenada de manera anónima con el fin de tener información en un futuro para un algoritmo de recomendación de tipo filtrado colaborativo ítem-ítem, siendo el uso de este algoritmo opcional en el proceso de implementación.

4.4 VISTA FÍSICA

La vista física describe el sistema desde el punto de vista de un ingeniero de sistemas. Está relacionada con la topología de componentes de software en la capa física, así como las conexiones físicas entre los componentes. Para su representación, se empleará un diagrama de despliegue sobre la aplicación web a desarrollar, el cual es mostrado a continuación:

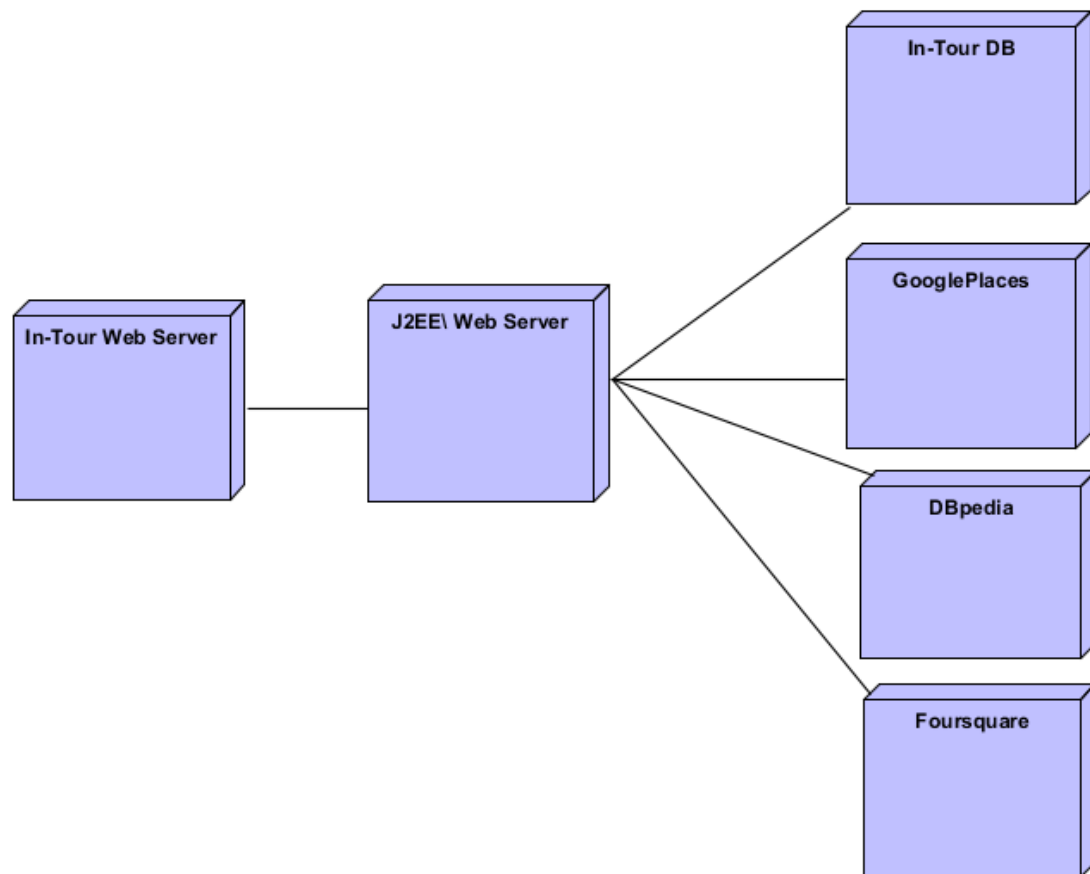


Ilustración 24: Diagrama de despliegue

En base al diagrama anterior, es necesario mencionar que esto ha implicado modelar la topología del hardware sobre el que se ejecuta el sistema. De este modo, este tipo de diagramas se ha diseñado para modelar muchos de los aspectos hardware de un sistema a un nivel suficiente para que se pueda especificar la plataforma sobre la que se ejecuta el software en el mismo.

Como en este caso se trata de un sistema con arquitectura *Cliente-Servidor*, es necesario realizar este tipo de diagrama ya que estos tipos de sistemas requieren tomar decisiones sobre la conectividad de red de los clientes a los servidores y sobre la distribución física de los componentes software del sistema a través de nodos.

Por consiguiente, en el caso de la aplicación *In-Tour Web* los nodos asociados a su diseño físico son aquellos que aparecen en el diagrama, los cuales son los siguientes:

- ***In-Tour Web Server***: Es el nodo que está formado por la interfaz a través de la cual el cliente interactúa con el sistema con el fin de obtener diferentes resultados y también por el conjunto de servicios web de los cuales hace uso la capa de presentación para comunicarse con la aplicación web. De este modo, se encuentra relacionado con el nodo *J2EE\ Web Server*.
- ***J2EE\ Web Server***: Se trata del nodo que representa el servidor web de la aplicación y que permite suministrar a los servicios web la respuesta solicitada. Por tanto, actúa de intermediario entre las peticiones realizadas a través de los servicios web desde el nodo *In-Tour Web Server* y los servicios externos, que en este caso se tratan de las bases de datos empleadas:
 - In-Tour DB.
 - Google Places.
 - DBpedia.
 - Foursquare.

Además, en dicho nodo se realizan todas las operaciones que constituyen a la aplicación web sirviéndose de las consultas a las bases de datos mencionadas con el fin de confeccionar el resultado esperado.

- ***In-Tour DB***: Nodo que representa la base de datos propia de la aplicación web y donde se almacenan las valoraciones realizadas por parte de los usuarios de las rutas generadas.
- ***GooglePlaces***: Nodo que representa la base de datos Google Places.
- ***DBpedia***: Nodo que representa la base de datos DBpedia.
- ***Foursquare***: Nodo que representa la base de datos Foursquare.

4.5 VISTA +1: ESCENARIOS

La vista de escenarios constituye la descripción de la arquitectura, la cual se ilustra utilizando un conjunto de casos de uso o escenarios lo que genera una quinta vista. Los escenarios van a describir secuencias de interacciones entre objetos, y entre procesos. Son empleados para identificar y validar el diseño de arquitectura además de servir de punto de partida para pruebas de un prototipo de arquitectura.

De este modo, esta vista va a tener la función principal de unir y relacionar las otras 4 vistas. Esto quiere decir que desde un caso de uso se puede observar cómo se van ligando las otras 4 vistas, con lo que tendremos una trazabilidad de componentes, clases, equipos, paquetes, etc..., para realizar cada caso de uso.

Por tanto, para la especificación de dicha vista es necesario mencionar que los casos de uso definidos en el apartado 3.2. *Casos de Uso* serán los mismos que para dicho apartado, de modo que se mantienen los mismos casos de uso tanto para la fase de análisis como para la de diseño.

4.6 ENTORNO TECNOLÓGICO SELECCIONADO

En el presente apartado se lleva a cabo una descripción de las diferentes tecnologías empleadas en el proyecto, las cuales han servido como medio para llevar a cabo el proceso de implementación de la aplicación web y, además, se mencionarán las razones por las que fueron seleccionadas.

4.6.1 Java

[10] Java es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems. Se trata de un lenguaje de programación con propósito general, concurrente y orientado a objetos, que fue diseñado con el fin de tener pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, es decir, que el código que es ejecutado en una plataforma no tiene que ser recompilado para ser ejecutado en otra.

Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones cliente-servidor de web, con unos 10 millones de usuarios reportados. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

Por consiguiente, el lenguaje fue creado con cinco objetivos principales:

- Debería usar el paradigma de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.



Ilustración 25: Logo Java [10]

En definitiva, la razón por la que dicha tecnología fue seleccionada para el proyecto fue debido a que Java permite desarrollar, implementar y utilizar de forma eficaz las aplicaciones y servicios, además de ser empleado desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet. A continuación, se mostrarán los porcentajes de utilización del mismo según la página oficial de Java:

- *El 97% de los escritorios empresariales ejecutan Java.*
- *El 89% de los escritorios (o computadoras) en Estados Unidos ejecutan Java.*
- *9 millones de desarrolladores de Java en todo el mundo.*
- *La primera opción para los desarrolladores.*
- *La primera plataforma de desarrollo.*

- *3 mil millones de teléfonos móviles ejecutan Java.*
- *El 100% de los reproductores de Blu-ray incluyen Java.*
- *5 mil millones de Java Cards en uso.*
- *125 millones de dispositivos de televisión ejecutan Java.*
- *5 de los 5 principales fabricantes de equipos originales utilizan Java ME.*

De este modo, a continuación se mencionarán los aspectos que nos aporta emplear dicha plataforma:

- Escribir software en una plataforma y ejecutarla virtualmente en otra.
- Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles.
- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML y mucho más.
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización.
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, microcontroladores, módulos inalámbricos, sensores, gateways, productos de consumo y prácticamente cualquier otro dispositivo electrónico.

4.6.2 Servicio Web: REST

[11] Respecto a dicha tecnología empleada, es necesario mencionar en primer lugar la definición de un servicio web. De este modo, un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. Es una máquina que atiende a las peticiones de los clientes web y les envía los recursos solicitados.

La principal razón para usar servicios web es que se pueden utilizar con HTTP sobre TCP en el puerto 80. Dado que las organizaciones protegen sus redes mediante firewalls que filtran y bloquean gran parte del tráfico de Internet, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios web utilizan este puerto, por la simple razón de que no resultan bloqueados. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el más común.

Otra razón es que, antes de que existiera SOAP, no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red. Las que había eran ad hoc y poco conocidas, tales como EDI(Electronic Data Interchange), RPC (Remote Procedure Call), u otras APIs.

Una tercera razón por la que los servicios web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.

En este caso, el tipo de servicio web el cual será empleado será REST, el cual es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. Dicha tecnología es empleada generalmente para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON...etc.) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como por ejemplo SOAP.

Por consiguiente, este tipo de servicio web ha sido seleccionado para la realización del proyecto debido a que desde su creación la web ha disfrutado de escalabilidad debido a una serie de diseños fundamentales que son mencionados a continuación:



Ilustración 26: Logo WebServices REST [11]

- Un protocolo cliente/servidor sin estado: Este aspecto permite que ni el cliente ni el servidor necesiten recordar ningún estado de las comunicaciones entre mensajes.
- Operaciones bien definidas: Dichas operaciones se aplican a todos los recursos de información. Las más importantes: POST, GET, PUT y DELETE.
- Una sintaxis universal para identificar los recursos.
- El uso de hipermedios, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente HTML o XML.

4.6.3 Maven

[12] Maven se trata de una herramienta software que será empleada para la gestión y construcción del proyecto y además puede ser empleado al implementarse mediante el lenguaje de programación Java.

De este modo, dicha herramienta utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

Una característica clave de Maven es que está listo para usar en red. El motor incluido en su núcleo puede dinámicamente descargar plugins de un repositorio, el mismo repositorio que provee acceso a muchas versiones de diferentes proyectos Open Source en Java, de Apache y otras organizaciones y desarrolladores. Además, Maven provee soporte no solo para obtener archivos de su repositorio, sino también para subir artefactos al repositorio al final de la construcción de la aplicación, dejándola al acceso de todos los usuarios. Una caché local de artefactos actúa como la primera fuente para sincronizar la salida de los proyectos a un sistema local.

Por consiguiente, esta herramienta está construida alrededor de la idea de reutilización, y más específicamente, a la reutilización de la lógica de construcción. Como los proyectos generalmente se construyen en patrones similares, una elección lógica podría ser reutilizar los procesos de construcción.

Aunque Maven es configurable, históricamente el proyecto Maven ha enfatizado seriamente que los usuarios deben adherirse a su concepto de un modelo de proyecto estándar tanto como sea posible.



Ilustración 27: Logo Maven [12]

Las partes del ciclo de vida del proyecto Maven que han sido empleadas son las siguientes:

- **compile:** Genera los ficheros .class compilando las fuentes .java.
- **test:** Ejecuta los test automáticos de JUnit existentes, abortando el proceso si alguno de ellos falla.
- **package:** Genera el fichero .jar con los .class compilados.
- **install:** Copia el fichero .jar a un directorio de nuestro ordenador donde maven deja todos los .jar. De esta forma esos .jar pueden utilizarse en otros proyectos maven en el mismo ordenador.
- **deploy:** Copia el fichero .jar a un servidor remoto, poniéndolo disponible para cualquier proyecto maven con acceso a ese servidor remoto.

4.6.4 Git

[13] Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.



Ilustración 28: Logo Git [13]

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o frontend. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Resultó de la experiencia del diseñador de Linux, Linus Torvalds, manteniendo una enorme cantidad de código distribuida y gestionada por mucha gente, incide en numerosos detalles de rendimiento, y de la necesidad de rapidez en una primera implementación.

Por consiguiente, a continuación se mencionaran aquellas características que resultan beneficiosas para emplearlas en el desarrollo del proyecto:

- Fuerte apoyo al desarrollo no lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones. Git incluye herramientas específicas para navegar y visualizar un historial de desarrollo no lineal. Por tanto, un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente, conforme se pasa entre varios programadores que lo revisan.
- Gestión distribuida: Git le proporciona a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.
- Los almacenes de información pueden publicarse por HTTP, FTP, rsync o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de cifrado SSH.
- Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.
- Todas las versiones previas a un cambio determinado, implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio (denominado autenticación criptográfica de historial).
- Los renombrados se trabajan basándose en similitudes entre ficheros. Aparte de nombres de ficheros, pero no se hacen marcas explícitas de cambios de nombre con base en supuestos nombres únicos de nodos de sistema de ficheros, lo que evita posibles, y posiblemente desastrosas, coincidencias de ficheros diferentes en un único nombre.
- Realmacenamiento periódico en paquetes (ficheros). Esto es relativamente eficiente para escritura de cambios y relativamente ineficiente para lectura si el reempaquetado (con base en diferencias) no ocurre cada cierto tiempo.

Como dicha tecnología es proporcionada por la plataforma de desarrollo github, la misma será empleada para realizar un control de versiones del proyecto.

5 PRUEBAS REALIZADAS

Las pruebas de software consisten en la dinámica de la verificación del comportamiento de un programa mediante un conjunto finito de casos de prueba, con el objetivo de proporcionar información objetiva e independiente de la calidad del producto a la parte interesada o Stakeholder.

Por consiguiente, son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa, probando el comportamiento del mismo.

De este modo, en el presente apartado se mostrarán las diferentes pruebas realizadas con el fin de verificar aquellos requisitos definidos en el apartado 3.3.5 *Especificación de Requisitos* y así permitir comprobar el correcto funcionamiento del proyecto y el cumplimiento de los objetivos establecidos en el inicio del mismo.

A continuación, se muestra la plantilla empleada para determinar los diferentes casos de prueba:

Identificador: TIPO-XX	
Título	
Descripción	
Resultado esperado	
Dependencias	

Tabla 70: Plantilla de pruebas

Donde:

- **Identificador:** Permite identificar unívocamente cada prueba en función de la codificación TIPO-XX:
 - TIPO: Se refiere al tipo de prueba. En el caso de las pruebas unitarias tomará el valor UNI, de las pruebas de integración el valor INT y de las pruebas de usabilidad y accesibilidad el valor USL.
 - XX: Hace referencia al número identificador de la prueba.
- **Título:** Título asignado a la prueba.
- **Descripción:** Determina cómo será llevada a cabo la prueba y su objetivo al aplicarla.
- **Resultado esperado:** Indica el resultado que se espera que sea obtenido tras la aplicación de la prueba.
- **Dependencias:** Hace referencia a los requisitos relacionados a partir de los cuales se decidió realizar la prueba.

5.1 PRUEBAS UNITARIAS

Identificador: UNI-01	
Título	Tipo de entidad buscada
Descripción	Una vez que el usuario ha realizado la búsqueda por entidad, al mismo tiempo se comprobará el tipo de dato buscado en cada base de datos accedida.
Resultado esperado	El tipo de dato obtenido se encontrará entre los siguientes: <ul style="list-style-type: none">• PERSONA.• LUGAR.• ORGANIZACIÓN.• EVENTO.• OTROS.
Dependencias	RSF-01

Tabla 71: Prueba unitaria UNI-01

Identificador: UNI-02	
Título	Número de resultados en DBpedia
Descripción	Se calculará el número de resultados obtenidos a partir de las consultas realizadas en la base de datos DBpedia.
Resultado esperado	El número de características obtenidas a partir de la búsqueda por entidad mediante DBpedia se encontrará en el intervalo [0,7].
Dependencias	RSF-02

Tabla 72: Prueba unitaria UNI-02

Identificador: UNI-03	
Título	Número de resultados en Google Places
Descripción	Se calculará el número de resultados obtenidos a partir de las consultas realizadas en la base de datos Google Places.
Resultado esperado	El número de características obtenidas a partir de la búsqueda por entidad mediante Google Places se encontrará en el intervalo [0,5].
Dependencias	RSF-03

Tabla 73: Prueba unitaria UNI-03

Identificador: UNI-04	
Título	Características de Ruta
Descripción	Se comprobará que la longitud del array asociado a los aspectos seleccionados por parte del usuario sea igual o mayor a 2.
Resultado esperado	Se podrá proceder a generar la ruta.
Dependencias	RSF-09

Tabla 74: Prueba unitaria UNI-04

Identificador: UNI-05	
Título	Localizaciones de características
Descripción	Se comprobará que el número de POIs en la ruta es igual al número de características seleccionadas para generar la ruta.
Resultado esperado	A cada característica le corresponderá una localización o POI.
Dependencias	RSF-11

Tabla 75: Prueba unitaria UNI-05

Identificador: UNI-06	
Título	Guardar valoraciones
Descripción	Se comprobará que se ha almacenado correctamente la valoración de cada ruta si el mensaje de respuesta tras dicho proceso ha sido OK.
Resultado esperado	La ruta será almacenada correctamente y el mensaje de respuesta será OK.
Dependencias	RSF-14

Tabla 76: Prueba unitaria UNI-06

Identificador: UNI-07	
Título	Radio de POIs alrededor
Descripción	Se comprobará que el radio de distancia de cada POI obtenido alrededor de una determinada localización toma un valor máximo de 10000 metros.
Resultado esperado	Los puntos de interés obtenidos alrededor de una localización dada no se encontrarán más lejos de 10000 metros respecto a la misma.
Dependencias	RSF-15

Tabla 77: Prueba unitaria UNI-07

Identificador: UNI-08	
Título	Número de POIs alrededor
Descripción	Se comprobará que el número de POIs obtenido alrededor de una determinada localización toma un valor máximo de 5.
Resultado esperado	Los puntos de interés obtenidos alrededor de una localización dada serán como máximo 5.
Dependencias	RSF-16

5.2 PRUEBAS DE INTEGRACIÓN

Identificador: INT-01	
Título	Resultados obtenidos
Descripción	Una vez se ha realizado la búsqueda por entidad, se comprobará que la longitud del array asociado a las características obtenidas sea mayor a 0 en el caso de que la búsqueda haya sido satisfactoria.
Resultado esperado	El número de características mostradas en la interfaz serán mayor a 0 si la búsqueda ha sido satisfactoria.
Dependencias	RSF-04

Tabla 78: Prueba de integración INT-01

Identificador: INT-02	
Título	Aspectos seleccionados
Descripción	Se comprobará que las características seleccionadas por parte del usuario se han añadido en el array asociado a dichos aspectos seleccionados con el fin de no ocasionar ningún error a la hora de generar posteriormente la ruta.
Resultado esperado	En el caso de que el usuario haya seleccionado características a través de la interfaz, la longitud del array asociado a dichos aspectos seleccionados será mayor a 0.
Dependencias	RSF-05

Tabla 79: Prueba de integración INT-02

Identificador: INT-03	
Título	Aspectos seleccionados 2
Descripción	Se comprobará que las características seleccionadas han sido eliminadas en el array asociado a los resultados obtenidos en caso de que el usuario seleccione determinadas características a través de la interfaz.
Resultado esperado	Se eliminarán los elementos seleccionados del array asociado a los resultados obtenidos.
Dependencias	RSF-06

Tabla 80: Prueba de integración INT-03

Identificador: INT-04	
Título	Aspectos descartados
Descripción	Se comprobará que las características descartadas han sido añadidas en el array asociado a los resultados obtenidos en caso de que el usuario descarte determinadas características a través de la interfaz.
Resultado esperado	Se eliminarán los elementos seleccionados del array asociado a los resultados obtenidos.
Dependencias	RSF-07

Tabla 81: Prueba de integración INT-04

Identificador: INT-05	
Título	Aspectos descartados 2
Descripción	Se comprobará que las características descartadas han sido eliminadas en el array asociado a los aspectos seleccionados en caso de que el usuario descarte determinadas características a través de la interfaz.
Resultado esperado	Se eliminarán los elementos seleccionados del array asociado a los aspectos seleccionados.
Dependencias	RSF-08

Tabla 82: Prueba de integración INT-05

Identificador: INT-06	
Título	Combinación de características
Descripción	Se comprobará con qué entidad tiene relación cada característica.
Resultado esperado	Se podrán seleccionar características procedentes de diferentes entidades buscadas para generar la ruta.
Dependencias	RSF-10

Tabla 83: Prueba de integración INT-06

Identificador: INT-07	
Título	Markers del mapa
Descripción	Se comprobará que la longitud del array asociado a los markers en el mapa es igual al número de POIs pertenecientes a la ruta generada.
Resultado esperado	El número de POIs pertenecientes a la ruta, los cuales serán mostrados en orden en la interfaz corresponderán con el número de markers en el mapa.
Dependencias	RSF-12

Tabla 84: Prueba de integración INT-07

Identificador: INT-08	
Título	Información de POI
Descripción	Se comprobará que todos los aspectos que constituyen la información de cada POI sea distinto de vacío.
Resultado esperado	En el mapa al hacer click sobre cada marker aparecerán todos los campos informativos rellenos.
Dependencias	RSF-13

Tabla 85: Prueba de integración INT-08

5.3 PRUEBAS DE USABILIDAD Y ACCESIBILIDAD

Identificador: USL-01	
Título	Diseño Responsive
Descripción	La aplicación será ejecutada desde distintos tipos de dispositivos: ordenador, tablet y dispositivo móvil.
Resultado esperado	La interfaz será adaptativa a cualquier tipo de dispositivo.
Dependencias	RSNF-01

Tabla 86: Prueba de usabilidad y accesibilidad USL-01

Identificador: USL-02	
Título	Aprendizaje de la aplicación web
Descripción	Se reunirá a un grupo de personas para que hagan uso de la aplicación web y se calculará el tiempo medio de aprendizaje de la misma.
Resultado esperado	El tiempo medio de aprendizaje será menor a 1 hora.
Dependencias	RSNF-02

Tabla 87: Prueba de usabilidad y accesibilidad USL-02

Identificador: USL-03	
Título	Evaluación de la usabilidad
Descripción	Se realizará un análisis experto de todos los elementos de usabilidad de la aplicación web empleando las heurísticas de Nielsen
Resultado esperado	La interfaz será amigable para el usuario.
Dependencias	RSNF-03, RSNF-04, RSNF-06

Tabla 88: Prueba de usabilidad y accesibilidad USL-03

Identificador: USL-04	
Título	Evaluación de errores
Descripción	Los archivos HTML y CSSs de la aplicación web serán verificados mediante el test de validación de W3C.
Resultado esperado	La tasa de errores será menor al 1%.
Dependencias	RSNF-05

Tabla 89: Prueba de usabilidad y accesibilidad USL-04

5.4 RESULTADO DE LAS PRUEBAS.

En referencia a los resultados obtenidos tras la aplicación de las pruebas, es necesario mencionar que fueron comprobadas tras finalizar el proceso de implementación de la misma y se obtuvo lo siguiente:

PRUEBA	RESULTADO
UNI-01	ÉXITO
UNI-02	ÉXITO
UNI-03	ÉXITO
UNI-04	ÉXITO
UNI-05	ÉXITO
UNI-06	ÉXITO
UNI-07	ÉXITO
UNI-08	ÉXITO
INT-01	ÉXITO
INT-02	ÉXITO
INT-03	ÉXITO
INT-04	ÉXITO
INT-05	ÉXITO
INT-06	ÉXITO
INT-07	ÉXITO
INT-08	ÉXITO
USL-01	ÉXITO
USL-02	ÉXITO
USL-03	ÉXITO
USL-04	ÉXITO

Tabla 90: Resultados de pruebas

5.5 MÉTRICAS DE CÓDIGO

[14] La calidad del software, en cuanto a código fuente, es uno de los requisitos que se debe exigir a cualquier aplicación. Normalmente, no se plantea como uno de los requisitos funcionales prioritarios y esto provoca que aunque se haya realizado un buen análisis e incluso un buen diseño el código generado no es lo suficientemente bueno y es el origen de todos los posibles errores, y en consecuencia no se cumple con los requisitos funcionales establecidos. De este modo, la calidad de forma genérica tiene una definición más amplia:

- *Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.* R.S. Pressman (1992).
- *El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas.* ISO B402 (UNE 66-001-92).

Las definiciones de calidad anteriores se refieren especialmente a la metodología utilizada y al grado de satisfacción de los requisitos funcionales del producto software.

A continuación, se presentan unas métricas las cuales son medidas de calidad del código fuente, ya que se estiman que los requisitos funcionales han sido cubiertos al obtener un producto que ha sido probado a través de distintos conjuntos de pruebas y obteniendo unos tiempos de ejecución aceptables (siempre menor a 1 segundo).

Es necesario mencionar que dichas métricas de la calidad del código fuente han sido obtenidas mediante un plugin de Eclipse [15] que permite generar una serie de medidas significativas basadas en las definiciones realizadas en el libro *Object-Oriented Metrics, measures of Complexity*, de Brian Henderson-Sellers, Prentice Hall, 1996 .

Por consiguiente, éstas métricas miden, principalmente, los valores de cohesión y acoplamiento que nos permiten establecer si el código está bien estructurado:

- *Cohesión*: La cohesión es deseable debido a que una unidad (componente) representa una parte simple de solución. Si es necesario cambiar el sistema, la parte correspondiente está en un solo lugar y lo que se desee hacer con él estará encapsulado en él. La meta según Lawrence, es hacer que los componentes sean lo más cohesivos posible.
- *Acoplamiento*: Está relacionado con la cohesión. Es un indicador de la fuerza de interconexión entre los componentes o elementos de la arquitectura. Los sistemas altamente acoplados tienen una fuerte interconexión, lo que se refleja en una gran dependencia entre los componentes. Los sistemas poco acoplados, por otro lado, tienen poca relación entre sus componentes o elementos. El objetivo, según Lawrence, es mantener el acoplamiento en el nivel más bajo posible, la conectividad sencilla entre módulos da como resultado un software que es más fácil de comprender y menos propenso al “efecto onda” producido cuando los errores aparecen en una posición y se propagan a lo largo del sistema.

De este modo, en primer lugar será necesario indicar aquellas medidas que se van a tener en cuenta:

- Number of static Methods (NSM): Número total de métodos estáticos.
- Total Lines of Code (TLOC): Número total de líneas de código. No incluye comentarios ni líneas en blanco.
- Afferent Coupling (CA): Número de clases fuera del paquete que dependen de clases dentro del paquete.
- Normalized Distance (RMD): Este número debe ser pequeño. Cercano a 0 indica un buen diseño de paquetes.
- Number of Classes (NOC): Número de clases.
- Specialization Index (SIX): Índice de especialización. Se calcula de la siguiente manera: $NORM \cdot DIT / NOM$.
- Instability (RMI): La inestabilidad. Se calcula de la siguiente manera: $CE / (CA + CE)$.
- Number of Atributtes (NOF): Número de atributos definidos en un ámbito.
- Number of Package (NOP): Número total de paquetes.
- Method lines of Code (MLOC): Número de líneas de código por método. No incluye ni comentarios ni espacios en blanco.
- Weighted methods per Class (WMC): Suma de las complejidades ciclomáticas de McCabe de todos los métodos de una clase.
- Number of Overriden Methods (NORM): Número total de métodos sobrescritos. No incluye los de la clase "Object".
- Number of Static Attributes (NSF): Número total de atributos estáticos.
- Nested Block Depth (NBD)
- Number of methods (NOM): Número de métodos definidos en un ámbito.
- Lack of Cohesion of Methods (LCOM): Cohesión de una clase. Un valor bajo indica clase con alta cohesión (preferible) y alto, baja cohesión y se debería dividir en n subclases.
- McCabe Cyclomatic Complexity (VG): Complejidad ciclomática.
- Number of Parameters (PAR): Número de parámetros.
- Abstractness (RMA): Número de clases e interfaces abstractas en un paquete dividido entre el total de tipos en el paquete.
- Number of Interfaces (NOI): Número de interfaces.
- Efferent Coupling (CE): Número de clases dentro del paquete que dependen de clases fuera del paquete.
- Depth of Inheritance Tree (DIT): Distancia de una clase en la jerarquía de herencia.

Por consiguiente, los resultados obtenidos tras hacer uso de la herramienta en Eclipse, han sido los siguientes:

MÉTRICA	TOTAL	MEDIA	DESVIACIÓN TÍPICA	RANGO	MÁXIMO
PAR		0.778	0.737	5	2
NSF	9	0.529	0.776		2
CE		1.1	0.7		2
SIX		0.004	0.015		0.062
NOC	17	1.7	0.781		3
NOF	14	0.824	1.653		7
RMA		0.09	0.181		0.5
RMD		0.253	0.335		1
NSM	19	1.118	1.843		7
NOI	3	0.3	0.64		2
TLOC	583				
WMC	72	4.235	4.278		16
NOM	35	2.059	3.718		16
DIT		1.176	0.706		4
NOP	10				
RMI		0.69	0.347		1
VG		1.333	0.839	10	6
NBD		1.333	0.667	5	4
LCOM		0.09	0.248		0.857
MLOC	224	4.148	5.635		25
NORM	1	0.059	0.235		1
CA		1.4	2.107		7

Tabla 91: Métricas de la calidad del código

Según los rangos establecidos por defecto para cada medida en la herramienta empleada, se ha podido observar que los resultados anteriormente obtenidos se encuentran dentro de los rangos fijados, tanto del propio código como del código generado. Aunque sobre el código generado no se tiene control de generación pero si es necesario asegurar que la herramienta está generando buen código, ya que de otra forma repercutiría de manera negativa a los módulos propios. Además, de esta forma también se pueden valorar las herramientas de generación de código con el objeto de utilizar una u otra.

Sobre los valores obtenidos hay que destacar los siguientes puntos:

- La complejidad ciclomática de McCabe (VG) es siempre aceptable, obteniendo una media de 1.333 por debajo del rango con valor 10, y con un máximo de 6 en una clase.
- Respecto a LCOM o medida de cohesión de una clase se obtiene una media de 0.09. En esta medida el valor adecuado es cercano a 0, por lo que se puede concluir que las clases del proyecto tienen una excelente valoración en cohesión. El valor máximo es de 0.857 que se da en la clase *POI*, siendo este valor coherente ya que almacena y concentra mucha información.
- Las medidas de acoplamiento CA y CE de nuevo dan buenos valores para todas las clases.
- Los valores obtenidos respecto a la métrica PAR, con una media de 0.778 por debajo del rango 5 y con un máximo de 2, tomando la mayoría de las clases los valores 0 y 1.

Con estas medidas, se ha creado un producto software cuyo código posee las características deseadas de alta cohesión y bajo acoplamiento. No obstante, toda aplicación web y por extensión su código fuente son siempre candidatos a mejorar.

6 PLANIFICACIÓN Y PRESUPUESTO

La planificación de proyectos es importante ya que no puede funcionar sin establecer los plazos, la entrega y la disponibilidad de los recursos del mismo, ya se trate de personal, inventario o capital. Por esta razón, cualquier proyecto software sin una determinada organización de las tareas es uno condenado a encontrar problemas en el ciclo de vida del mismo.

El objetivo de la planificación de un proyecto software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos costosos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo del mismo, y deberían actualizarse regularmente a medida que progresa el proyecto.

De este modo, a continuación se mostrará la planificación realizada para dicho proyecto y la cuál se ha seguido para la obtención del mismo y después se especificará una estimación de los costes que ha supuesto mostrándolo a través de un presupuesto realizado.

6.1 PLANIFICACIÓN DEL PROYECTO

Para la planificación del proyecto se empleará la herramienta *Ganttproject* la cual nos permitirá construir un diagrama de Gantt.

El diagrama de Gantt es una útil herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. Por tanto, a continuación de mostraran las diferentes tareas que se han tenido en cuenta durante la realización del proyecto:

- 1. Planteamiento del proyecto y reuniones.**
 - Estudio del contenido.
 - Reuniones.
- 2. Formación en el uso de DBpedia.**
 - Búsqueda de información sobre DBpedia.
 - Aprendizaje de lenguaje SPARQL (lenguaje para el acceso a la base de datos DBpedia).
 - Generación de pequeñas consultas sobre la interfaz gráfica de DBpedia que permitía ejecutarlas.
- 3. Definición de objetivos.**
- 4. Estado del Arte.**
 - Búsqueda de información sobre aplicaciones similares.
 - Análisis comparativo y selección de funcionalidades principales del proyecto.
- 5. Análisis**
 - Requisitos de usuario.
 - Casos de uso.
 - Requisitos Software.
- 6. Diseño**
 - Vista lógica.
 - Vista de desarrollo.
 - Vista de procesos.
 - Vista física.
 - Vista de escenarios.

7. Implementación

- Construcción de consultas en el lenguaje SPARQL para la búsqueda por entidad.
- Desarrollo del buscador por entidad a través de DBpedia.
- Construcción de consultas en el lenguaje SPARQL para obtener localizaciones a través de características.
- Desarrollo de generador de rutas inicial.
- Mejora de la interfaz de usuario.
- Implementación de la selección y descarte de características.
- Desarrollo de la 2ª versión del generador de rutas.
- Construcción de consultas realizadas a las bases de datos Google Places para la búsqueda por entidad.
- Inclusión al buscador de entidad consultas realizadas a la base de datos Google Places.
- Construcción de consultas para obtener localizaciones a través de las características mediante las bases de datos Google Places y Foursquare.
- Desarrollo de la 3ª versión del generador de rutas.
- Desarrollo de la valoración de rutas.
- Desarrollo de la obtención y visualización de información asociada a cada POI.
- 2ª mejora de la interfaz de usuario.

8. Realización de Pruebas.

9. Realización de la documentación del proyecto (El presente documento).

Tras mencionar aquellas tareas que se han tenido en cuenta a la hora de realizar la planificación del proyecto, se ha obtenido el siguiente diagrama de Gantt:

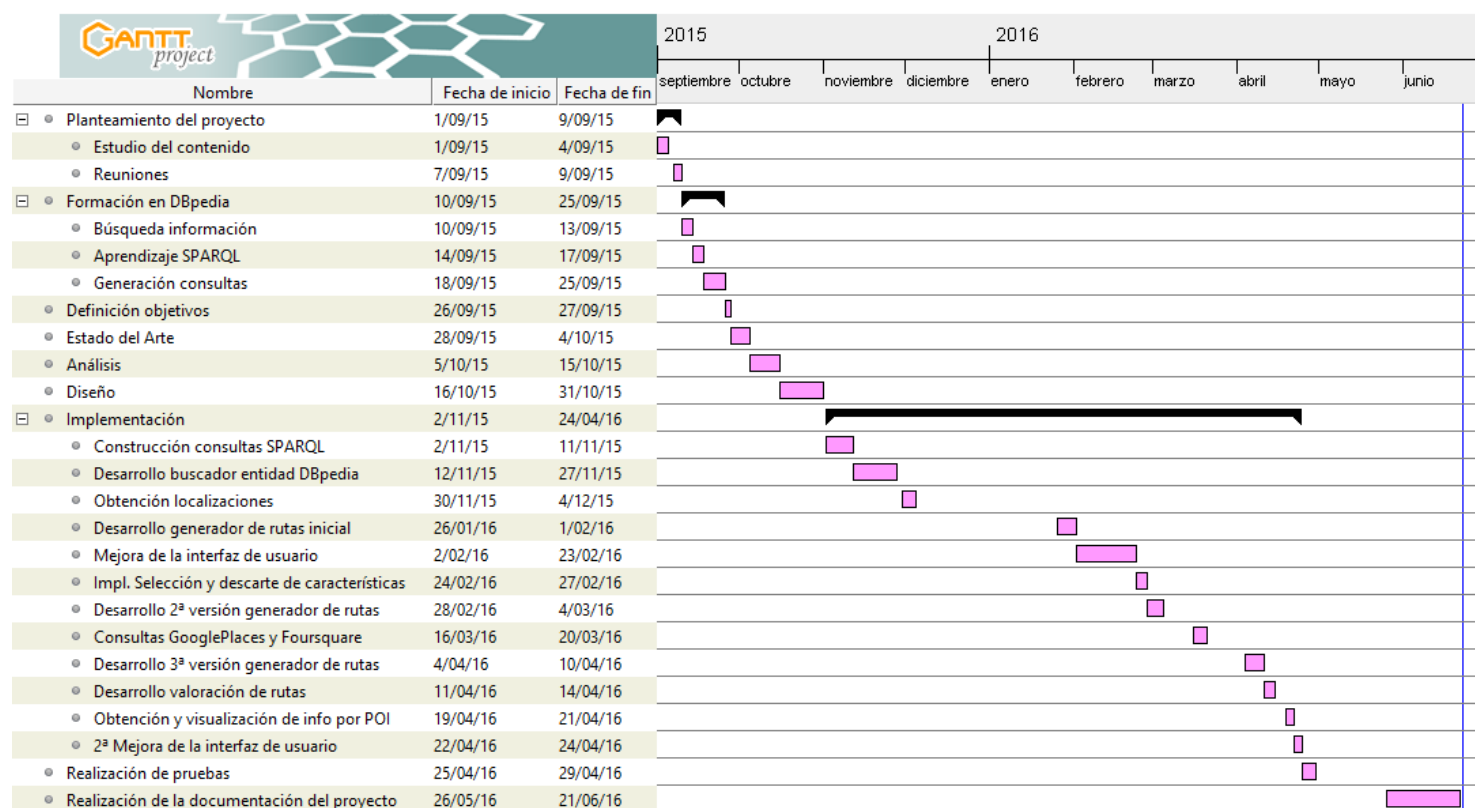


Ilustración 29: Diagrama de Gantt

Como bien podemos observar en el anterior diagrama, el proyecto tuvo su inicio el día 01/09/2015 y concluyó el día 21/06/2016 dando lugar a un total de 268 días dedicados al mismo y de 427 horas aproximadamente. Es necesario mencionar que se han cumplido todos los plazos establecidos en la planificación con el fin de realizar las tareas, sin embargo, en cuanto a las horas cabe destacar que inicialmente se estimó un total de 300 horas pero debido a diferentes problemas causados preferentemente en el proceso de implementación no se ha dado una desviación en cuanto a las fechas, pero sí en cuanto al número de horas empleadas con un aumento del 42.33% del esfuerzo dedicado.

A continuación se mostrará una tabla que recogerá las principales tareas realizadas junto con la asignación de los días, las horas estimadas y las horas finalmente empleadas en cada una de ellas:

Nº DE TAREA	TAREA	Nº DE DÍAS EMPLEADOS	Nº DE HORAS ESTIMADAS	Nº DE HORAS EMPLEADAS
1	Planteamiento del proyecto	9	5	5
2	Formación en DBpedia	16	28	30
3	Definición de objetivos	2	3	3
4	Estado del Arte	7	5	10
5	Análisis	11	13	26
6	Diseño	16	25	37
7	Implementación	175	186	246
8	Realización de pruebas	5	5	11
9	Realización de la documentación del proyecto	27	30	59
TOTAL:		268	300	427

Tabla 92: Planificación de horas y días

Es necesario mencionar y, como bien se ha comentado antes, no se ha producido ninguna desviación de las fechas estimadas, esto ha sido debido a que las tareas han sido organizadas en función de aquellos días en las que se tuviera mayor disponibilidad y además en función de aquellos meses que son periodos de exámenes como Diciembre, Enero y Mayo, que tal y como se muestra en el diagrama en dichos meses no se asignó ninguna tarea a realizar.

En cuanto a las tareas críticas, cabe destacar *Formación en DBpedia* ya sin un aprendizaje previo del lenguaje SPARQL no hubiera sido posible realizar las consultas a dicha base de datos e implementar parte del proyecto. Por otro lado, otras tareas consideradas críticas son: *Desarrollo del generador de rutas* y *Construcción de consultas para acceder a las bases de datos Google Places y Foursquare*, ya que su realización es fundamental puesto que son las funciones principales del proyecto. Sin embargo, el resto de tareas se han considerado relevantes pero no críticas para la obtención de la aplicación web.

Por otro lado, es necesario comentar que la realización de la presentación no ha sido tomada en cuenta en el proceso de planificación debido a que será realizada fuera del plazo de la entrega de la presente documentación del proyecto.

6.2 PRESUPUESTO DEL PROYECTO

Tras comentar aquellos aspectos en relación a la planificación de las tareas que se tuvieron que realizar para la confección del proyecto, es necesario mencionar también los gastos que ha supuesto. Para ello, a continuación se realizará una estimación del coste total de la realización del mismo teniendo en cuenta los gastos de personal, coste del material empleado y los costes indirectos.

6.2.1 Costes de personal

Respecto al coste del personal es necesario mencionar que aquellos implicados son los siguientes:

- *Marta Plaza Rodríguez*: Desarrolladora de la aplicación web y autora del presente documento.
- *José María Álvarez Rodríguez*: Tutor del proyecto.

Sin embargo, con el tutor del proyecto se han llevado a cabo una serie de reuniones con el fin de solventar determinadas dudas y proporcionar ayuda en el desarrollo de dicho trabajo fin de grado. De esta manera, no se ha considerado un coste personal como tal, por tanto, solamente se tendrá en cuenta los costes que supone la autora del mismo.

Tal y como se mostró en la *Tabla 92: Planificación de horas y días*, se mencionaron el total de horas estimadas y aquellas empleadas realmente. De este modo, teniendo en cuenta aquellas que finalmente fueron utilizadas se han obtenidos los siguientes gastos de personal:

PERSONAL	CARGO	SALARIO o COSTE POR HORA (€/h)	Nº TOTAL DE HORAS EMPLEADAS (h)	COSTE (€)
Marta Plaza Rodríguez	Informático Junior	25	427	10675
TOTAL:				10.675

Tabla 93: Costes de personal

El coste que ha implicado el personal mostrado en la tabla anterior se ha calculado estimando una media de 25 €/h ya que como el proyecto no ha sido realizado de una manera continua cada semana y cada mes, se ha considerado más apropiado realizar una estimación por cada hora empleada. Esto permitirá, en el caso de que el mismo fuera llevado a cabo mediante una empresa, aproximar mejor el coste de personal que supondría en base a la jornada de trabajo de cada empleado.

6.2.2 Costes del material

La realización del proyecto se ha llevado a cabo en un ordenador portátil propio de la autora del presente documento, el cual tiene las siguientes características:

- Fabricante: Packard Bell
- Modelo: EasyNote TM86
- Procesador: Intel (R) Core(TM) i3 CPU M 330 @ 2,13GHz 2,13 GHz
- Memoria instalada (RAM): 4,00 GB (3,68 utilizable)
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64.
- Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para la pantalla.

En base a las características mencionadas, el precio del portátil es contribuable al coste total del proyecto, suponiendo un coste de aproximadamente 749 € y, considerando que el periodo medio de amortización de un ordenador es de 48 meses, el gasto proporcional al proyecto será:

$$749\text{€} / 48 \text{ meses} = 16,54\text{€/ mes}$$

Y teniendo en cuenta que se ha hecho uso del mismo aproximadamente durante 10 meses esto dará un total de:

$$16,54\text{€/ mes} \cdot 10 \text{ meses} = 165,41 \text{ €}$$

TOTAL COSTE HARDWARE: 165,41 €

Por otro lado, se tendrá que tener en cuenta las licencias del sistema operativo empleado, el cual ha sido la versión básica *Windows 10 Home* con un coste de 135 € además de la licencia de *Microsoft Office Professional 2013* con un coste de 65,34 €. Por tanto, si tenemos en cuenta el porcentaje de utilización de dicho material software el precio se reduce:

- *Windows 10 Home* ha sido empleado aproximadamente durante 10 meses y considerando que el periodo medio de amortización para dicho sistema operativo es de 36 meses, el gasto que ha implicado durante la realización del proyecto ha sido de:

$$135 \text{ €} / 36 \text{ meses} = 3,75 \text{ €} / \text{mes}$$

$$3,75 \text{ €} \cdot 10 \text{ meses} = 37,5 \text{ €}$$

- *Microsoft Office Professional 2013* ha sido empleado aproximadamente durante 10 meses y considerando que el periodo medio de amortización para dicho sistema operativo es de 36 meses también, el gasto que ha implicado durante la realización del proyecto ha sido de:

$$65,34 \text{ €} / 36 \text{ meses} = 1,82 \text{ €} / \text{mes}$$

$$1,82 \text{ €} \cdot 2 \text{ meses} = 3,64 \text{ €}$$

Por tanto, esto dará lugar a un coste total en software de:

$$37,5 \text{ €} + 3,64 \text{ €} = 41,14 \text{ €}$$

TOTAL COSTE SOFTWARE: 41,14 €

A continuación se mostrará una tabla que resume aquello mencionado:

MATERIAL	TIPO	PRECIO (€)	Nº DE MESES EMPLEADO	PERÍODO DE AMORTIZACIÓN (meses)	COSTE (€)
Ordenador portátil Packard Bell EasyNote TM86	Hardware	749	10	48	165,41
Windows 10 Home	Software	135	10	36	37,5
Microsoft Office Professional 2013	Software	65,34	10	36	3,64
TOTAL:					206,55

Tabla 94: Costes del material

6.2.3 Costes indirectos

En cuanto a los costes indirectos se ha tenido en cuenta lo siguiente:

COSTE INDIRECTO	COSTE MENSUAL (€)	Nº DE MESES EMPLEADO	COSTE (€)
ADSL y luz	40	10	400
Transporte del domicilio a la universidad	84	4	336
TOTAL:			736

Tabla 95: Costes indirectos

Respecto a los costes indirectos es necesario mencionar que en referencia al transporte del domicilio a la Universidad ha sido necesario incluirlo ya que durante 4 meses la autora tuvo que realizar el proyecto en la Universidad a causa de tener que realizar las prácticas de otras asignaturas al mismo tiempo.

6.2.4 Coste total del proyecto

A continuación, se mostrará una tabla en la que se recogen los distintos costes mencionados y que se han tenido en cuenta para realizar el presupuesto del proyecto:

TIPO DE COSTE	COSTE (€)
COSTES DE PERSONAL	10.675
COSTES DEL MATERIAL	206,55
COSTES INDIRECTOS	736
IMPORTE TOTAL SIN BENEFICIO	11.617,55
BENEFICIO ESPERADO (15%)	1.742,63
IMPORTE TOTAL CON BENEFICIO	13.360,18
IVA (21%)	2.805,63
TOTAL:	16.165,81

Tabla 96: Coste Total del Proyecto

Como podemos observar en la tabla anterior, el coste total del proyecto supondrá **16.165,81€** (Dieciséis mil ciento sesenta y cinco euros con ochenta y un céntimos).

7 ÉTICA Y MARCO LEGAL

Respecto al marco legal tenido en cuenta durante la realización del proyecto, se ha contemplado por un lado el marco regulatorio para el uso mediante terceros de la información que obra en el sector turístico a través de los servicios externos empleados y el marco regulatorio para el uso de la información empleada en la propia aplicación web.

Respecto al uso de la información a través de terceros, es necesario mencionar que al emplearse servicios externos a partir de los cuales se han obtenido los datos necesarios para realizar la implementación de la aplicación web, es preciso comentar aquellos determinados aspectos considerados con el fin de mantener la legalidad y moralidad en toda acción realizada:

- **DBpedia:** Se trata de una infraestructura de datos públicos, es decir, es un repositorio en la web, abierto y gratuito, con información estructurada proveniente de Wikipedia. El acceso a dicha base de datos está disponible bajo los términos de:
 - *La licencia Creative Commons Attribution-ShareAlike 3.0*, en la cual se indica que aquel que haga uso de DBpedia podrá copiar y redistribuir el material en cualquier medio o formato, transformar y aprovechar el material para cualquier fin, incluso comercial no pudiendo el usuario de la misma revocar estas libertades mientras siga los términos de la licencia [16].
 - *La licencia GNU Free Documentation* [17].
- **Google Places:** Permite el registro gratuito de las empresas que hago uso de la misma en los mapas de Google (Google Maps), para que de esta forma cuando los usuarios de Google hagan búsquedas locales puedan encontrar aquellas registradas visualizando su ubicación, teléfono, horario de atención, fotos, etc. Para hacer uso de dicha API es necesario obtener una clave de manera gratuita, sin embargo, tendrá un servicio más restringido permitiendo un máximo de 150000 peticiones en el mismo día. Si se deseara disponer de un mejor servicio que soportara mayor número de solicitudes, se tendría que pagar por ello. En este caso se ha hecho uso de la versión gratuita de acceso y cumpliendo la normativa vigente en [18].
- **Foursquare:** Se trata de un servicio basado en localización web que permite principalmente a las empresas promocionar su negocio y a sus usuarios encontrar lugares para comer, beber, visitar o comprar en cualquier ciudad del mundo. Para hacer uso de dicha API es necesario al igual que en Google Places disponer de unos determinados credenciales para hacer uso de la versión gratuita y a la vez ocurriendo el mismo caso que anteriormente, es decir, se da un número limitado de peticiones por día y si se deseara hacer uso de un mejor servicio se tendrían que pagar los costes correspondientes. En este caso, no se ha considerado necesario aumentar el número de solicitudes disponibles por día ya que es más que suficiente para el desarrollo del trabajo fin de grado. Por tanto, se ha hecho uso de la versión gratuita de Foursquare cumpliendo todos los términos y condiciones indicados en el acuerdo expuesto y en la *Política de la plataforma de Foursquare*, todo ello indicado en [19].

Ante lo comentado anteriormente, se afirma que todos los recursos y datos de los que se ha hecho uso a través de servicios externos de terceros en la aplicación web desarrollada ha sido cumpliendo la normativa y las leyes pertinentes a cada licencia adquirida al acceder al acuerdo con dichos servicios.

Por otro lado, respecto a los datos obtenidos a partir de la propia aplicación, es necesario mencionar aquellas leyes tenidas en cuenta:

- **LOPD (*Ley de Protección de Datos de Carácter Personal*)**: La protección de datos es un derecho fundamental reconocido por la Constitución y regulado por la Ley Orgánica 15/1999, de 13 de Diciembre, de Protección de Datos de Carácter Personal (LOPD), así como por el Reglamento de Desarrollo de la misma, aprobado por Real Decreto 1720/2007, de 21 de Diciembre (RDLOPD). Por tanto, dicha ley tiene como objetivo garantizar y proteger, en lo que concierne al tratamiento de los datos personales, de libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar. Dicha información ha sido extraída de [20].

8 ENTORNO SOCIO-ECONÓMICO

[21]El sector turístico es un área de actividad que utiliza de forma intensiva tanto la información como las transacciones en todas y cada una de las etapas de su cadena de valor. Un sector con una enorme y variada oferta y con una demanda potencial que se puede cuantificar en miles de millones de personas precisa de la existencia de grandes sistemas de intermediación, que antes eran de carácter físico pero que, con la llegada de Internet, han sufrido una transformación radical.

Esta transformación ha supuesto la desaparición de numerosos agentes y el nacimiento de otros, en muchas ocasiones procedentes del mundo tecnológico. Las innovaciones introducidas en los servicios y la creación de nuevos modelos de negocio han provocado la transformación de la cadena de valor en su conjunto. La nube, la movilidad, lo social... son elementos imprescindibles a lo largo de toda la cadena de valor, entre otras razones porque los usuarios y las empresas las consideran elementos de simplificación de los procesos y de ahorro de costes. Además, hay que sumar las redes sociales, donde los turistas comparten sus experiencias, y la economía colaborativa, en la que juegan un papel esencial los contenidos y las prescripciones realizadas por los usuarios, antes, durante y después del viaje.

En esta revolución hay empresas que aprovechan las tecnologías para innovar y ofrecer nuevos productos, servicios y modelos de negocio turísticos a sus clientes; otras las incorporan para reinventarse mejorando sus procesos y la calidad de sus productos. Un tercer grupo de empresas observa los cambios con preocupación, al no ser capaz de tomar medidas para no quedarse rezagado. En el mundo del turismo la competencia se basa ya en la presencia, diferenciación y reputación online, referentes en la estrategia de una empresa turística digital.

El objetivo final es crear valor y experiencias atractivas para el usuario. Este, por su parte, emplea la tecnología en un ciclo que comienza con la búsqueda de información, sigue con la selección y comparación de precios y experiencias, se extiende a la compra, que cada vez se realiza más de forma móvil, y finaliza con la compartición e intercambio de información, lo que sirve para retroalimentar el ciclo.

Una diferencia importante con escenarios anteriores del sector turístico es que los viajeros digitales siguen buscando información y contratando servicios durante el camino a su destino y también una vez en el mismo, gracias a que suelen llevar permanentemente encima uno o varios dispositivos móviles, especialmente en el caso del colectivo de turistas más jóvenes. El resultado es una demanda que requiere atención 24x7 y en la que las reservas de última hora tienen un peso cada vez más importante.

Según la Organización Mundial de Turismo, España, con unos ingresos por turismo internacional por valor aproximado de 57.866 millones de euros, ocupa la primera posición en Europa y la segunda mundial.

En España, según los datos recogidos en el Informe de perspectivas turísticas realizado anualmente por Exceltur¹¹ correspondiente a 2015, el turismo ha sido el principal sector generador de renta y empleo en España, habiendo contribuido con 0,5 puntos de crecimiento adicional del PIB nacional. El sector turístico aporta 124.000 millones de euros, lo que representa ya el 11,7% del total del PIB Español.

En términos de empleo, según la EPA 2016, el sector turístico y de viajes representa 2,32 millones de puestos de trabajo, tanto directos como indirectos. Viajes y Turismo emplea

directamente más del doble de los puestos de trabajo de los que genera, por ejemplo, el sector de servicios financieros.

Según lo mencionado, a continuación se muestra un gráfico sobre el crecimiento anual del PIB turístico (ISTE) comparado con el PIB general de la economía española desde el año 2006 al 2015.

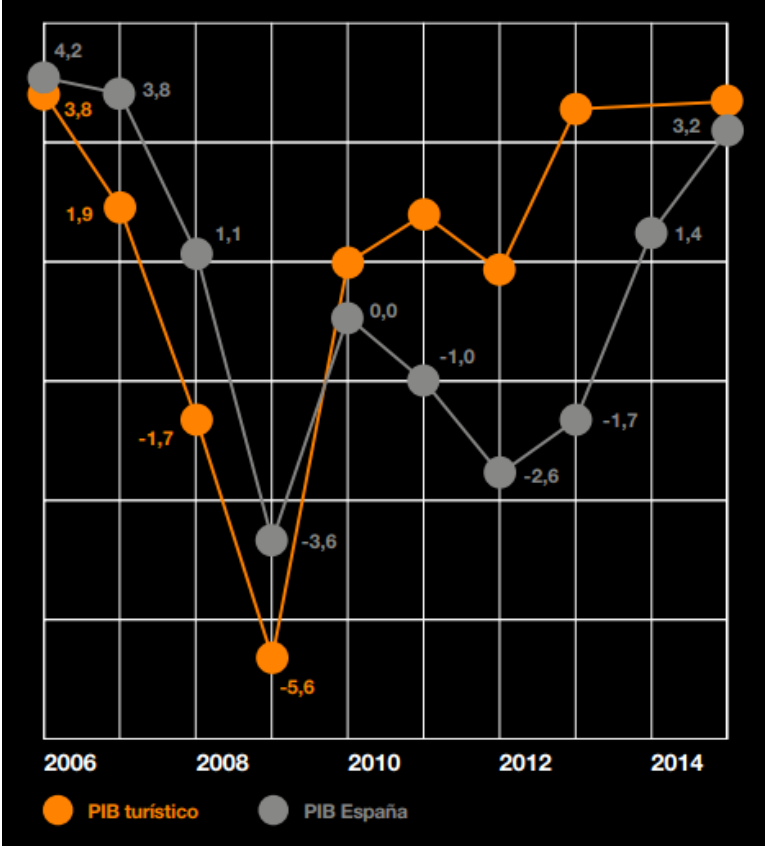


Ilustración 30: Crecimiento anual del PIB turístico [21]

Además, podemos observar los ingresos obtenidos en el turismo internacional del año 2014, donde se puede apreciar que España se encuentra en el segundo puesto más alto:

	\$ EEUU			
	Miles de millones		Variación %	
	2013	2014	13/12	14/13
Estados Unidos	172,9	177,2	7,0	2,5
España	62,6	65,2	7,6	4,2
China	51,7	56,9	3,3	10,2
Francia	56,7	55,4	5,6	-2,3
Macao (China)	51,8	50,8	18,1	-1,9
Italia	43,9	45,5	6,6	3,7
Reino Unido	41,0	45,3	12,1	10,3
Alemania	41,3	43,3	8,2	5,3
Tailandia	41,8	38,4	23,4	-8,0
Hong Kong (China)	38,9	38,4	17,7	-1,4

Ilustración 31: Ingresos por turismo interacional 2014 [21]

Por consiguiente, la conclusión que puede ser extraída en base a aquello mencionado anteriormente, es que el turismo se trata de un sector industrial el cual es un motor de economías de muchos países y en España tal y como se ha comentado es muy importante, por tanto, las TIC serán un factor clave y críticas para mejorar los servicios del mismo.

9 CONCLUSIONES Y TRABAJO FUTURO

El sistema diseñado y desarrollado, es el inicio de un futuro motor de recomendaciones que ayudará al usuario a planificar de manera inteligente y dinámica las rutas que desea realizar durante su viaje.

Su desarrollo surgió de la necesidad de una estrategia y de un impulso en el sector turístico, debido a que se evidencian cambios cada vez mayores, a causa del comportamiento y preferencias de los viajeros, ya que hoy en día se encuentran cada vez más informados y conocen más de los diferentes destinos y sus dinámicas, no solo en el ámbito nacional sino internacional, debido a que actualmente las fronteras se traspasan sin límites.

De este modo, la creación de dicho proyecto pretende brindar al viajero la oportunidad de gestionar su viaje de la forma más rápida y sencilla posible, por tanto, ha sido una satisfacción llevar a cabo el mismo con el fin de complacer e incentivar que la actividad turística sea alcanzable por cualquier tipo de usuario a la vez de darle la posibilidad de adquirir conocimientos de interés sobre cada lugar visitado perteneciente a las rutas generadas.

El principal objetivo del mismo era implementar una aplicación web que mostrara al usuario metadatos pertenecientes a una palabra (ya fuera una persona, lugar, organización, evento u otros) introducida por el mismo en el buscador y a partir de dichos metadatos obtener puntos de interés que permitieran generar rutas que el usuario pudiera completar. Para ello, finalmente se ha obtenido un sistema capacitado para:

- *Buscar por entidad:* El sistema permite no sólo buscar por una localización, sino que proporciona la posibilidad de buscar por lugar, persona, organización o evento, entre otros. Como resultado de la búsqueda, se obtendrá un conjunto de características las cuales podrán ser seleccionadas o descartadas por el usuario. Esto permite ampliar el área de búsqueda para planificar el viaje deseado, a diferencia de las aplicaciones web similares que fueron analizadas que no incluyen dicha funcionalidad.
- *Añadir/Eliminar características:* El sistema permite que el usuario añada o elimine aquellas características que estén asociadas a la entidad buscada (persona, lugar, organización, evento u otros) y también podrán seleccionarse otras características obtenidas a partir de búsquedas anteriores. El objetivo en este caso, será realizar una combinación de características procedentes de diferentes búsquedas con el fin de obtener determinados conocimientos culturales, históricos,...etc. mientras el usuario visita los lugares que compongan la ruta generada a través de dichas características. En las aplicaciones web similares dicha acción no ha sido adoptada, ya que en las mismas solo se pueden añadir puntos de interés que necesariamente estén alrededor de un destino.
- *Generación de rutas automáticamente:* El sistema permite generar rutas realizando una planificación automática. De este modo, el usuario tiene la única tarea de añadir o eliminar aquellas características deseadas y la aplicación web automáticamente obtendrá una ruta compuesta por puntos de interés asociados a las características seleccionadas por parte del usuario, respectivamente. Dichos puntos serán devueltos al usuario de manera ordenada según el criterio de la distancia más corta entre los mismos, sin necesidad de que sea el usuario el que organice y planifique el orden de las visitas realizadas.

La presente funcionalidad se trata de uno de los puntos clave de la aplicación desarrollada, ya que en el caso de otras páginas web de turismo se dan los siguientes aspectos:

- El usuario solamente obtiene un único resultado cuando realiza búsquedas para un destino determinado.
- El usuario sólo podrá añadir a su planificación lugares que necesariamente se encuentren cercanos al destino buscado.
- El usuario solamente tiene la posibilidad de planificar manualmente la ruta turística que desee realizar durante su viaje.
- El orden de visita de cada localización deberá ser planificado por parte del usuario y no de manera automática.

De esta manera, esto se ha considerado un gran beneficio permitiendo reducir el esfuerzo empleado por parte del usuario en la organización de sus viajes de una manera sencilla.

- *Ubicación de localizaciones en el mapa:* El sistema mostrará en el mapa aquellos puntos de interés o POIs asociados a cada una de las características seleccionadas por parte del usuario para generar la ruta.
- *Mostrar información multimedia de cada POI:* El sistema permite que el usuario haga click sobre cualquier localización marcada en el mapa y se muestre la siguiente información:
 - Nombre de la característica de la que proviene el punto de interés.
 - Latitud.
 - Longitud.
 - Dirección del punto de interés.
 - Fuente de búsqueda de la que proviene: DBpedia, Google Places o Foursquare.
 - Imagen representativa del lugar.

A pesar de que dicha funcionalidad no es una novedad en las aplicaciones similares, se consideró de gran utilidad para identificar cada punto de interés.

- *Valoración de ruta generada:* El sistema permite valorar las rutas generadas sin necesidad de tener que valorar cada punto que constituya la ruta, a diferencia de la mayoría de las aplicaciones similares que dan la posibilidad de realizar una valoración sobre un único lugar visitado o en el caso de aquellas que tienen la posibilidad de generar rutas de manera manual, tampoco es posible valorar la ruta completa.

Respecto a las dificultades y problemas surgidos, es preciso mencionar que la mayoría de estos fueron la configuración de las librerías empleadas durante el proceso de comunicación con los servicios externos, es decir, las bases de datos de las que se hizo uso. Por otro lado, también surgieron diversos problemas durante la transferencia de datos enviados desde la capa de presentación al servidor de la aplicación. A pesar de ello, se consiguió resolverlos satisfactoriamente.

En definitiva, se ha desarrollado un sistema que ha cumplido todos los objetivos propuestos en sus inicios, pero también es necesario mencionar que el aspecto clave de la aplicación web es que se puede conectar a otros servicios de terceros, obtener información y de ahí realizar una mejor ordenación. Por tanto, el objetivo no era que dicho proyecto fuera mejor que las bases de datos empleadas (DBpedia, Google Places o Foursquare) pero sí se pretendía aunar más

información y también producir mejor información contextual capturando la misma a partir de distintas fuentes.

En base a lo comentado anteriormente, como todo sistema creado, puede ser mejorado de diversas maneras. Entre las mejoras posibles cabe destacar las siguientes:

- *Cumplimiento de logros*: Esta mejora consistirá en amoldar la aplicación como si de un juego se tratase pero en relación a la vida real. De este modo, en base a la aplicación web desarrollada, se podría incluir la concesión de logros tras cada ruta que fuera completada por el usuario. Esta funcionalidad añadida consistiría en que cada vez que el usuario llegara a un punto de interés perteneciente a la ruta generada, este tendría la opción de indicar que ha completado uno de los lugares de la ruta y, por tanto, se comprobaría comparando la posición geográfica del usuario mediante su geolocalización con la latitud y longitud del punto de interés de la ruta que supuestamente acaba de visitar.
- *Red social*: Esta mejora permitirá que los usuarios tengan un perfil en el que podrán mostrar aquellos viajes realizados mediante la publicación de fotos o vídeos, podrán hablar vía chat y aportar comentarios en cada ruta generada, entre otras acciones. Esta funcionalidad es empleada por la mayoría de las páginas web similares que fueron analizadas, por tanto, se ha considerado un aspecto importante a incluir.
- *Implementación en iOS y Android*: En el caso de que la aplicación fuera empleada por un número considerable de usuarios, sería un aspecto importante desarrollar el funcionamiento de *In-Tour Web* en una aplicación para dispositivos móviles, siendo más fáciles y cómodas de utilizar que una página web móvil permitiendo el aprovechamiento de las funcionalidades del dispositivo tales como la cámara o la geolocalización.
- *Coste de ruta*: Otra de las posibles mejoras que se podría incluir consistiría en que tras generar la ruta, se podría calcular el coste aproximado de lo que supondría realizar dicho viaje. Por tanto, esto permitirá al usuario elegir aquellas rutas más económicas en su planificación turística.
- *Recomendaciones*: Tal y como se comentó anteriormente, se puede hacer un uso eficiente de las valoraciones de rutas que fueron almacenadas. De este modo, cada valoración era almacenada de manera anónima, por tanto, esta mejora propuesta consistirá en hacer uso de dicha información la cual constituye los datos de entrada de un algoritmo de recomendación de tipo filtrado colaborativo ítem-ítem, para generar recomendaciones que sean mostradas al usuario y así ayudarle a tomar una mejor decisión sobre sus viajes.

10 BIBLIOGRAFÍA

- [1] «Revista Turismo & Tecnología,» Revista Turismo & Tecnología, [En línea]. Available: <http://www.turismoytecnologia.com/component/k2/item/1281-la-teor%C3%ADa-de-la-pir%C3%A1mide-de-maslow-aplicada-a-la-anal%C3%ADtica-web>.
- [2] Instituto de Estudios Turísticos, «Ministerio de Industria, Energía y Turismo,» 2003. [En línea]. Available: <http://estadisticas.tourspain.es/img-iet/Revistas/RET-158-2003-pag5-41-91949.pdf>.
- [3] «TripAdvisor,» [En línea]. Available: <https://www.tripadvisor.es/>.
- [4] «TouristEye,» [En línea]. Available: <http://www.touristeye.es/>.
- [5] «Minube,» [En línea]. Available: <http://www.minube.com/>.
- [6] «TripIt,» [En línea]. Available: <https://www.tripit.com/>.
- [7] «The Culture Trip,» [En línea]. Available: <http://theculturetrip.com/>.
- [8] «UCI, School of Information & Computer Science,» Noviembre 1995. [En línea]. Available: <http://www.ics.uci.edu/~andre/ics223w2006/kruchten3.pdf>.
- [9] Escuela Ingeniería Industrial, «Pontificia Universidad Católica Valparaíso,» [En línea]. Available: <http://ocw.pucv.cl/cursos-1/arquitectura-de-sistemas-de-software/materiales-de-clases/web-cliente-servidor>.
- [10] «Java Web Site,» [En línea]. Available: <https://www.java.com/es/about/>.
- [11] «The Java EE 6 Tutorial,» [En línea]. Available: <http://docs.oracle.com/javase/6/tutorial/doc/gijqy.html>.
- [12] «Maven Web Site,» 18 Junio 2016. [En línea]. Available: <https://maven.apache.org/what-is-maven.html>.
- [13] «Git Web Site,» [En línea]. Available: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>.
- [14] J. M. Á. Rodríguez, «SourceForge,» [En línea]. Available: <https://sourceforge.net/p/ontospread/code/HEAD/tree/trunk/pfc-doc-es/manual-tecnico/metricas/metricas.tex#l101>.
- [15] «Metrics SourceForge,» [En línea]. Available: <http://metrics.sourceforge.net/>.
- [16] «Creative Commons,» [En línea]. Available: <https://creativecommons.org/licenses/by-sa/3.0/>.
- [17] «GNU Operating System,» 12 Abril 2014. [En línea]. Available: <http://www.gnu.org/licenses/fdl-1.3.en.html>.

- [18] Google, «API de Google Maps,» [En línea]. Available:
<https://developers.google.com/maps/pricing-and-plans/?hl=es#details>.
- [19] Foursquare, «Foursquare,» 11 Abril 2014. [En línea]. Available:
<https://es.foursquare.com/legal/api/licenseagreement> .
- [20] Agencia Estatal Boletín Oficial del Estado, «LOPD,» [En línea]. Available:
<https://www.boe.es/buscar/act.php?id=BOE-A-1999-23750>.
- [21] Fundación Orange, Marzo 2016. [En línea]. Available:
http://www.thinktur.org/media/eE_La_transformacion_digital_del_sector_turistico.pdf.

11 GLOSARIO: ACRÓNIMOS, DEFINICIONES Y ABREVIATURAS

- *POI*: Hace referencia a una localización o punto de interés.
- *Localización*: Lugar en el que se localiza alguien o algo. En este caso se refiere a localización geográfica, formada por latitud y longitud o el nombre de un lugar.
- *Google Maps*: Es un servidor de aplicaciones de mapas en la web. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle.
- *RSF*: Requisito de software funcional.
- *RSNF*: Requisito de software no funcional.
- *RU*: Requisito de usuario.
- *CU*: Caso de uso.
- *Front-end*: Representa la capa de presentación de la aplicación web, es decir, representa la información de la misma al usuario.
- *Aplicación web*: Se trata de una herramienta que los usuarios emplean accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.
- *UNI*: Prueba unitaria.
- *INT*: Prueba de integración.
- *USL*: Prueba de usabilidad y accesibilidad.
- *Rango*: Representa la diferencia entre el valor máximo y el valor mínimo de un conjunto de datos.
- *Stakeholder*: Es el término en inglés que define a los interesados en un proyecto de desarrollo de software.
- *PIB*: Producto interior bruto. Es una magnitud macroeconómica que expresa el valor monetario de la producción de bienes y servicios de demanda final de un país.
- *Marker*: Icono que se emplea de apuntador en el caso de Google Maps.
- *Brainstorming*: Es una herramienta de trabajo grupal que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado.